2009-12

# The classification of e-authentication protocols for targeted applicability

Chia, Wan Yin.

Monterey, California. Naval Postgraduate School


# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**THE CLASSIFICATION OF E-AUTHENTICATION PROTOCOLS FOR TARGETED APPLICABILITY**

by

Wan Yin Chia

December 2009

Thesis Co-Advisors: J. D. Fulp
Ted Huffmire

THIS PAGE INTENTIONALLY LEFT BLANK

| REPORT DOCUMENTATION PAGE | | Form Approved OMB No. 0704-0188 |
|---|---|---|



| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE<br>December 2009 | 3. REPORT TYPE AND DATES COVERED<br>Master's Thesis |
|---|---|---|

| 4. TITLE AND SUBTITLE<br>The Classification of E-Authentication Protocols for Targeted Applicability | 5. FUNDING NUMBERS |
|---|---|
| 6. AUTHOR<br>Wan Yin Chia | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)<br>Naval Postgraduate School<br>Monterey, CA 93943-5000 | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)<br>N/A | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER |

**11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

| 12a. DISTRIBUTION / AVAILABILITY STATEMENT<br>Approved for public release; distribution is unlimited | 12b. DISTRIBUTION CODE |
|---|---|

**13. ABSTRACT**

Authentication is a fundamental aspect of information security in enabling the authenticity of the source of information to be determined. Among several electronic authentication mechanisms available today, deploying the right authentication mechanism will protect information against its envisaged threat(s) in the designated operating environment. This study attempts to create a taxonomy (classification) for current operational authentication protocols, and show how the taxonomy could help to determine the appropriate protocol to meet a particular operating environment's authentication needs. The approach used in this study's taxonomy development was to perform functional decomposition of the protocol in terms of the functionality it provides, the mechanisms it utilizes, and the key elements in facilitating its operation. This enabled a breaking-down into the fundamental building blocks of what makes up the protocol. The development of the taxonomy in this way enabled different perspectives and analyses of the protocols' capabilities and their applicability.

The basic idea of authentication via proof of possession of a secret, whether it is symmetric or asymmetric, applies for all categories of authentication protocols under study. Several use cases are put forth illustrating how the classification can be leveraged to facilitate analysis of the applicability of the protocol for implementation in a given targeted environment.

| 14. SUBJECT TERMS<br>Authentication Protocols, Taxonomy, Protocol Classification, Protocol Applicability, Operating Environment | 15. NUMBER OF PAGES<br>93 |
|---|---|
| | 16. PRICE CODE |

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UU |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release; distribution is unlimited**

# THE CLASSIFICATION OF E-AUTHENTICATION PROTOCOLS FOR TARGETED APPLICABILITY

Wan Yin Chia
Defence Science Technology Agency, Singapore
B.S., National University of Singapore, 1998

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN COMPUTER SCIENCE**

from the

**NAVAL POSTGRADUATE SCHOOL**
**December 2009**

Author: Wan Yin Chia

Approved by: J. D. Fulp
Thesis Co-Advisor

Ted Huffmire
Thesis Co-Advisor

Dr Peter Denning
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

Authentication is a fundamental aspect of information security in enabling the authenticity of the source of information to be determined. Among several electronic authentication mechanisms available today, deploying the right authentication mechanism will protect information against its envisaged threat(s) in the designated operating environment. This study attempts to create a taxonomy (classification) for current operational authentication protocols, and show how the taxonomy could help to determine the appropriate protocol to meet a particular operating environment's authentication needs. The approach used in this study's taxonomy development was to perform functional decomposition of the protocol in terms of the functionality it provides, the mechanisms it utilizes, and the key elements in facilitating its operation. This enabled a breaking-down into the fundamental building blocks of what makes up the protocol. The development of the taxonomy in this way enabled different perspectives and analyses of the protocols' capabilities and their applicability.

The basic idea of authentication via proof of possession of a secret, whether it is symmetric or asymmetric, applies for all categories of authentication protocols under study. Several use cases are put forth illustrating how the classification can be leveraged to facilitate analysis of the applicability of the protocol for implementation in a given targeted environment.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

# LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| AAA | Authentication, Authorization and Accounting |
| AES | Advanced Encryption Standard |
| CAPTCHA | Completely Automated Turing Test to tell Computers and Humans Apart |
| CHAP | Challenge Handshake Authentication Protocol |
| CIA | Confidentiality, Integrity, Availability |
| CRL | Certificate Revocation List |
| CSP | Credential Service Provider |
| DES | Data Encryption Standard |
| EAP | Extensible Authentication Protocol |
| GSM | Global System for Mobile Communication |
| GTC | Generic Token Card |
| HLR | Home Location Register |
| IKE | Internet Key Exchange |
| IMSI | International Mobile Subscriber Identity |
| ISAKMP | Internet Security Architecture Key Management Protocol |
| KDC | Key Distribution Center |
| LM | Lan Manager |
| MAC | Message Authentication Code |
| NIST | National Institute of Standards and Technology |
| OTP | One-Time Password |
| PAP | Password Authentication Protocol |
| PEAP | Protected EAP |
| PII | Personal Identifiable Information |
| PoP | Proof of Possession |
| PPP | Point-to-Point Protocol |
| RADIUS | Remote Access Dial-in User Service |
| RFID | Radio Frequency Identification Technology |
| RP | Relying Party |
| SA | Security Association |

| | |
|---|---|
| SAML | Security Assertions Markup Language |
| SIM | Subscriber Identity Module |
| SMS | Short Message Service |
| SPI | Security Parameter Index |
| SRP | Secure Remote Password |
| SSH | Secure Shell |
| SSL/TLS | Secure Socket Layer/ Transport Layer Security |
| TACACS+ | Terminal Access Controller Access Control System Plus |
| TGT | Ticket Granting Ticket |
| TTLS | Tunneled TLS |
| VPN | Virtual Private Network |
| WEP | Wires Equivalent Privacy |
| WPA | Wi-Fi Protected Access |
| XML | Extensible Markup Language |

# EXECUTIVE SUMMARY

Authentication is a fundamental aspect of information security. In the current Information Age, it is crucial to be able to ascertain the authenticity of the source of information; that the originator of a unit of information is indeed as claimed. There are several electronic authentication mechanisms available today to protect information and provide identity management. Deploying the right authentication mechanism will target protection of information against its envisaged threat in the designated operating environment. Each authentication mechanism has its own strengths and weaknesses in terms of number of keys required to generate, cost, complexity, key distribution, security services provided, and vulnerabilities to certain threats such as spoofing and replay attack. There will be potential impacts in selecting one authentication mechanism over another for implementation in a particular domain. The security assurance level will also differ based on the selected authentication factor and underlying protocol mechanism.

An authentication protocol entails a sequence of messages exchanged between two parties, which allows the use/possession of some secret to be confirmed. It is almost certain that any authentication protocol will be dependent on parameters such as names and identities of authenticating parties, and any secrets shared between them. There are several authentication protocols and mechanisms available today. Each of these authentication protocols has some common mechanisms of performing authentication, though the implementation may differ in terms of strength and processes involved. The basic idea of authentication via proof of possession of secrets applies and remains the same for all categories of authentication protocols.

The authentication protocol key players examined here are an attempt to sample the various categories of authentication protocols available. These key players span from standard protocols for applications and network access, to

specific operating system protocols, to proprietary protocols. The focus will be on the underlying authentication *mechanism*, and the analysis will assume that the authenticating parties have already established all required prior configuration, certificate issuances, shared secret key agreement or key distribution requirements.

In the process of conducting the study of the various e-authentication protocols and developing the protocol taxonomy, the primary focus was on examining the mechanisms and key elements facilitating the authentication process. There may be differences in how each protocol is implemented; however, after peeling the outer layers and inspecting the underlying mechanism, it was determined that the fundamental mechanisms governing the way in which secrets are exchanged in an authentication session were common to all protocols. Proof of possession of a secret is conducted via asymmetric or symmetric means. Shared symmetric secret is the more commonly used means due to its efficiency, relative simplicity, and lower implementation cost. However, an asymmetric secret is necessary when non-repudiation is a required security service, and to support large-scale enterprises that are not conducive to dynamically establishing symmetric keys.

The basis of building the taxonomy is dependent on the application of the taxonomy. The approach used in this study's taxonomy development was to perform functional decomposition of the protocol in terms of the functionality it provides, the mechanisms it utilizes, and the key elements in facilitating the operation of protocol function. This enabled a breaking-down into the fundamental building blocks of what constitutes the fundamental authentication part of the protocol. The development of the taxonomy in this way enabled different perspectives and analyses of the protocols' capabilities and their applicability.

# ACKNOWLEDGMENTS

I would like to express my gratitude to all those who have helped and given me support one way or another to complete this thesis.

I owe my deepest gratitude to my thesis co-advisor, Mr J. D. Fulp, for his guidance, patience and support in this study. This thesis would not have been possible if not for his guidance and thought provoking ideas. I would also like to thank my thesis co-advisor, Dr Ted Huffmire, for his kind advice and support.

I am grateful to my sponsor Defence Science and Technology Agency (DSTA) for the opportunity and support they have given me.

Last but not least, special thanks to my family and friends in Singapore for their moral support given to me during my course of study in Monterey, NPS.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

## A. BACKGROUND

Authentication is a fundamental aspect of information security. In the current information age, the reliability and integrity of data is of great concern for organizations and individuals. It is crucial to be able to ascertain the authenticity of the source of information; i.e., that the originator of a unit of information is indeed as claimed. There are several electronic authentication mechanisms available today to protect information and provide identity management. Deploying the right authentication mechanism will protect information against its envisaged threat in the designated operating environment. This is especially challenging in the inter-networked, widely distributed, and open computing environment, which may involve remote authentication over the network, and with activities spanning from day-to-day email, business transactions over the web, to mobile wireless information exchanges.

Information security typically consists of the CIA (confidentiality integrity availability) triad. The confidentiality element is the protection of information to prevent unauthorized disclosure. The integrity element is the protection of information from unauthorized modification and ensuring data authenticity. It is the “I” element in the triad that is the focus of this study, and plays a significant role in establishing user identity and data integrity. Finally, the availability element refers to the information and service as being accessible to the user in a timely manner.

When referring to the “integrity” of information protection, it usually requires AAA (Authentication, Authorization and Accounting) support to work together to manage and control access to the protected information. Authentication refers to the identification process and serves to provide verification of the user’s identity. Authorization supports access control to ensure that the user is able to access only what he is "allowed" to. Accounting serves to

log user actions, information access, and information modification in support of audit security controls. These three processes are required to work together to ensure that the protection of information is complete, and ensure that the right user is allowed access to the information and that information modification is documented.

This thesis will focus on the authentication protocol in identity verification and protecting information integrity. For identity verification, this means the user needs to prove who he claims to be. The user may provide one or more pieces of evidence to prove his identity. The evidence may be in the form of a secret password shared between the user and the system, presentation of a valid identification card issued by a recognized authority, or some physiological characteristic based biometric that is bound to the user. Different authentication mechanisms can influence the assurance level of the protection of information. The authentication protocol is the implementation that leverages the authentication mechanism to facilitate secure data exchanges between communicating endpoints. As authentication may take place locally and remotely over the network, it is critical to consider the operating environment and network limitations to deploy the appropriate authentication protocol.

## B. WHAT IS E-AUTHENTICATION?

The definition of authentication is “the process of determining whether someone or something is, in fact, who or what it is declared to be” (SearchSecurity.com, 2007). Users who are required to be authenticated will have to *prove* their claimed identities.

E-authentication involves mechanisms to verify and ensure that only authorized users can log on to a particular domain and access data or network resources in an electronic manner. As such, users will be required to present their identity tokens electronically for validation. In some cases, electronic credentials are used during authentication. An electronic credential refers to a digital document or object that binds the user identity to the token possessed and

represents the user in gaining access to the information system locally as well as remotely. The fundamental components of an e-authentication infrastructure include registration, tokens, token management, authentication process and assertions (NIST,[1] 2008). The focus of this thesis is on the creation of a taxonomy of the authentication process and protocol.

### 1. Registration

Registration is the first step in e-authentication in which the user subscribes to some Registration Authority and is issued a secret token and a credential that binds to the user name by a Credential Service Provider (CSP). The token and credential may be used for subsequent authentication activities. The user is said to be a Subscriber of the CSP.

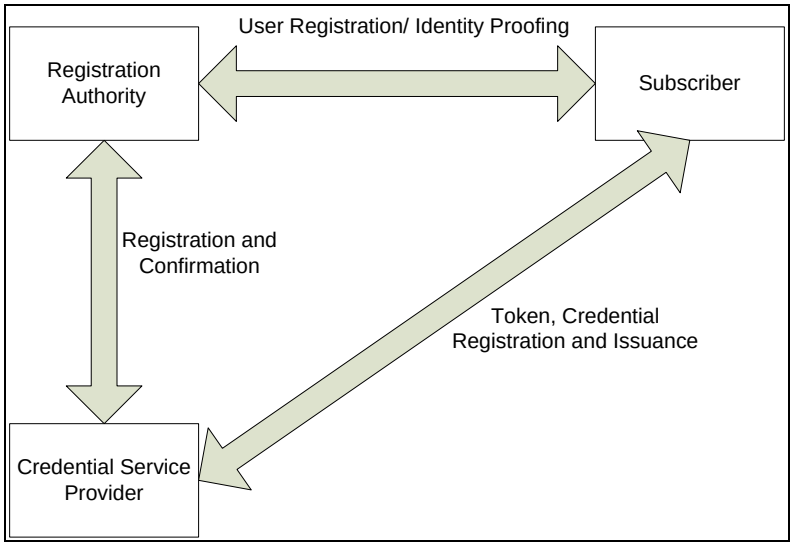


Figure 1. Registration Process (After NIST, 2008)

[1] National Institute of Standards and Technology (NIST) is the federal technology agency that develops and promotes measurements, standards and technology. http://www.nist.gov/index.html

## 2. Tokens

A token is what the user possesses to authenticate his identity. In general, each token may contain a secret or something that binds solely to the particular user. The three token types often considered are as follows:

a) What you know (e.g., password, private information of the user)

b) What you have (e.g., driver's license, ID smart card, cryptographic key, one-time password device)

c) What you are (biometric data such as fingerprints, hand writing metrics, facial features, or iris patterns)

A single token or a combination of two or more tokens (multi factor) may be used for authentication. The use of multi factor tokens typically enhances the assurance level of the overall authentication system.

## 3. Token and Credential Management

In general, the CSP is responsible for the token and credential management activities required to ensure the effectiveness of issuing of token and credential. The list of activities required is as follows:

a) Credential storage

   This is required for maintenance of credential storage whereby there should be protection against unauthorized modification. It will also need to be available for the CSP to perform verification of the token owner.

b) Token and credential verification

   The CSP is required to provide service to requested parties to facilitate a token and credential verification process.

c) Token and credential renewal

Tokens and credentials may be issued with limited life span or validity period. During the token renewal, the validity period is extended without changing the Subscriber's token and credential. However, in the event that the token type does not support the renewal process, a new token will be issued instead. If the credentials or tokens expire prior to the renewal process, the Subscriber may be required to go through the registration process again to re-establish his identity with the CSP.

d) Token and credential revocation

The CSP will need to maintain the revocation status of tokens and credentials. For example, public key certificates are revoked using a certificate revocation list (CRL). The CSP is responsible for maintaining an up to date CRL.

e) Token and credential destruction

This is required for the destruction of expired tokens and credential records. For credentials, it may be done through an update in the credential storage database. Some token types will need to be zeroized or destroyed to ensure all information pertaining to the Subscriber is deleted from the token, with no means of recovery.

## 4. Authentication Process

The user to be authenticated is usually called a Claimant, and the party verifying the identity is called a Verifier. The Claimant needs to prove to the Verifier that he possesses the token through an authentication protocol. The Verifier will then validate the token, possibly by interaction with the Claimant's Credential Service Provider, to confirm the Claimant identity. A Relying Party (RP) depends on the CSP or Verifier for the Claimant identity verification in order to process a transaction or grant access to some requested information, system or physical space. In some authentication environments, the RP may also serve

as the Verifier. The authentication process is usually facilitated by an authentication protocol. The authentication protocol entails the necessary data exchanges and processes that occur between the Claimant and Verifier, and it is these protocols that are the main interest of this taxonomy study.

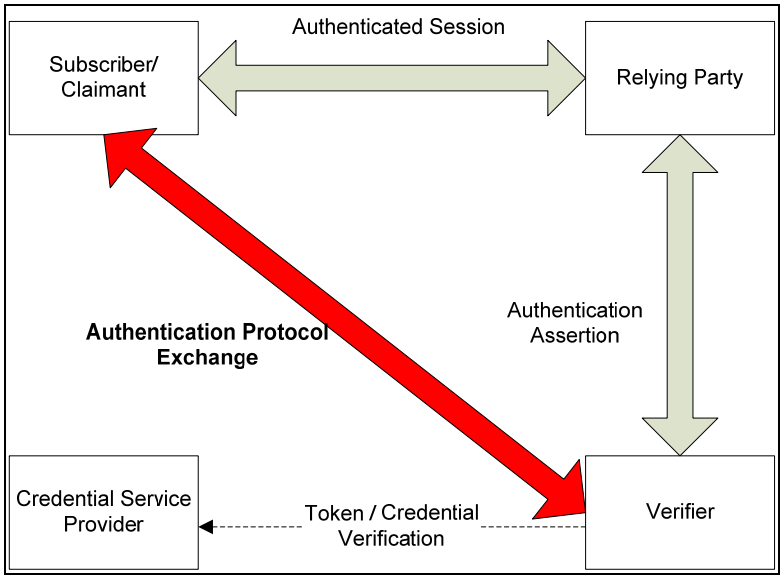


Figure 2. Authentication Process (After NIST, 2008)

## 5. Assertions

> Assertions are statements from a Verifier to a Relying Party that contain information about a Subscriber. Assertions are used when the Relying Party and the Verifier are not collocated. The Relying Party uses the information in the assertion to identify the Claimant, verify any presented/suggested identity attributes, and ultimately make authorization decisions regarding Claimant requests for any resources controlled by the Relying Party. (NIST, 2008)

Assertion mechanisms support federated identity management in which there could be multiple identity accounts held by the Subscriber with various Relying Parties. It supports single sign-on and facilitates authentication to be performed in lieu of, or in addition to, proof of identity from the Claimant.

SAML (Security Assertions Markup Language) is an XML (Extensible Markup Language) based framework for exchanging authentication and authorization data between security entities over the Internet.

## C. SCOPE AND OBJECTIVE

Several authentication technologies and mechanisms are available. The authentication protocols, just to name a few, include EAP (Extensible Authentication Protocol), CHAP (Challenge Handshake Authentication Protocol), Kerberos, RADIUS (Remote Access Dial-in User Service), and ISAKMP (Internet Security Architecture Key Management Protocol). Each of these authentication protocols is designed to provide a means of authentication for different usage scenarios and patterns. Each has its own strengths and weaknesses in terms of the number of keys required to generate, cost, complexity, key distribution method, security services provided, and vulnerabilities to certain threats such as spoofing and replay attacks. There will be potential security, cost, and scalability impacts in selecting one authentication mechanism over another for implementation in a particular domain. The assurance level provided will also differ based on the selected authentication factor and underlying protocol mechanism.

This research study attempts to create a taxonomy (classification) for currently operational authentication protocols, and show how the taxonomy can be leveraged to select appropriate protocols for specific authentication needs for a particular environment. It will provide emphasis on classification based on key features, functionality, strengths and potential vulnerabilities. Analysis will include how each protocol achieves or enhances the security objectives of integrity. Examples of use cases are evaluated to put forth how the classification can be leveraged to facilitate selection and applicability of the protocol for implementation in the targeted environment.

THIS PAGE INTENTIONALLY LEFT BLANK

# II. MECHANICS OF E-AUTHENTICATION PROTOCOLS

## A. INTRODUCTION

In the current Information Age, IT systems process and store a variety of information and provide access to a large number of users. The systems are accessible by different users who may not have access to all the information residing on them. Some information may be sensitive and should only be accessible by specific users. Logical access control to such information needs to be in place to control and monitor, by electronic means, the setting of permissions on files, folders and data; i.e. who can access what information. This differs from physical access control in which physical measures such as door locks restrict access to authorized personnel who have the key.

E-authentication protocols support and facilitate logical access control to information by performing the authentication process electronically to ascertain that the user is who he/she claims to be, so that subsequent access decisions based upon this can be made with greater confidence and assurance. The authentication protocol provides secure communication and data exchange between the authenticating parties (Claimant and Verifier/Relying Party) to establish the Claimant identity before the Claimant is granted access to the information resource.

The following sections describe the mechanics of a typical e-authentication protocol. In examining the mechanics of e-authentication protocol, the focus will be on the required message exchanges and transactions between the authenticating parties. In establishing the Claimant identity, it may be via presentation of proof of possession of a secret or some physiological trait (biometrics). Cryptographic techniques are employed to protect against disclosure of any authentication secret that may be conveyed between the authenticating parties. In some instances, authentication protocols may support

multifactor authentication, which typically results in higher assurance of the authentication. Lastly, some of the threats that authentication protocols will need to address are discussed.

## B. PROOF OF POSSESSION OF A SECRET

In order to authenticate and confirm the identity of a Claimant, presentation of proof of possession (PoP) of a secret is the most common mechanism. E-authentication is based on such proof, which may be implemented using a key, password, PIN, etc. The main essence of employing secret-based authentication is that the secret is only known to the Claimant and either the Verifier or Relying Party (V/RP), and serves as a form of identifier to the authenticating parties. In the case of shared secrets, the Verifier/Relying Party verifies and confirms the Claimant's *claim* to an identity based on the Claimant proving possession of some secret that has been pre-registered with the V/RP. It is assumed that both authenticating parties will protect the secret against unauthorized observation.

The protection of the secret is critical to prevent potential impersonation attacks. The secret is usually encrypted, or used to encrypt, or hashed when sent across networks to help maintain its secrecy. With such protection, the Verifier can be more confident that the Claimant is who he claims, if he is able to prove possession of this secret.

### 1. Symmetric vs. Asymmetric Secret

The secret used by the Claimant for authentication purposes can be a symmetric or asymmetric secret. In performing authentication exchanges between the authenticating parties, the symmetric secret is either used to encrypt a challenge or is hashed along with the challenge. Prior key agreement is required between the parties to decide on the symmetric key to use or how it can be derived for secure exchanges during authentication. It requires that the secret key remain secret at both sides. The symmetric secret can be static, e.g., a re-

used password, or it can be dynamic, e.g., a one-time password (OTP). The symmetric secret that is being exchanged may be stored in more than one place. It will be stored at least at the Claimant and Verifier's end, and perhaps at the Relying Party too. In light of the increased exposure of the symmetric secret and storage in multiple locations, it will need to be changed periodically if it is of the static form (e.g., password or PIN). A dynamic symmetric secret such as an OTP changes frequently by design.

An asymmetric secret mechanism involves a public and private key pair. The public key is publicly available, whereas the Claimant is the only one who has access to the corresponding private key. The keys comprising the pair are mathematically related such that data encrypted using one of the keys must be decrypted using the other key. Since the private key is kept in confidence by the Claimant, authentication exchanges can then be done to verify whether the public key can be used to decrypt any data encrypted by the Claimant's private key. The public key is made publicly available. In comparison to symmetric secret mechanisms, this reduces the risk of storing the secret at more than one place, which leads to less risk of divulging the authenticating secret.

### 2. Proof of Possession of Physiological Trait

Biometrics authentication refers to establishing identity based on physiological traits of a person such as their fingerprint, face, iris, voice, handwriting, or gait (Jain, 2006) (Gafurov, 2007). Compared to authentication based upon PoP of a secret, biometrics may appear to be more reliable, since biometric traits cannot be lost or forgotten. Biometrics are also able to provide for non-repudiation because of the difficulty of forgery of most biometrics and the uniqueness of appropriately chosen biometric traits.

Biometrics, however, are not secrets. It is generally "public knowledge" how a person may look or sound. For static presentation of biometrics, the Verifier will need to authenticate the Claimant using a biometric reader device where the Claimant is physically present and, ideally, observed as he/she

presents the biometric for measurement. It is not recommended to use static presentation of a biometric for remote authentication (NIST, 2008) in absence of a secure communication channel, to protect against the digitized biometric being "captured" and replayed later in an impersonation attack. Protection of biometric data is required during transmission and at the Verifier's storage end. This is to prevent replay and impersonation attacks. There is also a need to prevent proliferation and distribution of biometric data that may not be well regulated with respect to its linkage to other PII (personally identifiable information), and to manage the use of biometrics in authentication over remote networks (Wayman, 2008).

In consideration of biometrics used in conjunction with some form of challenge response mechanism, this will add some complexity to the authentication process as it results in a different authenticating response to each authentication transaction, rather than a specific digitized value that could be recorded for later use by an attacker. This will aid in the prevention of replay-based impersonation attacks. This would be appropriate for remote authentication with the correct response to the challenge effectively proving possession of the biometric being measured. In Australia, CentreLink provided a voice verification system to authenticate users for its call centre operations (Bingemann, 2009). Besides using voiceprints and pattern recognition software to recognize the speaker, it also incorporates additional authentication means such as allowing users to submit secret questions, and requesting users to recite a string of random numbers as part of the voice-verifying challenge-response exchange.

### 3. Multifactor Authentication

Multifactor authentication refers to the presentation of two or more authentication factors, such as a biometric used in conjunction with a password. It is also considered multifactor authentication if the authentication requires a token, the operation of which includes at least two authentication factors. For

example, a biometric is something you *are,* and a password is something you *know*. Multifactor authentication is often claimed to be more secure and able to meet more stringent security requirements than single factor authentication. However, authentication using multiple factors (e.g., biometric and password) may not necessarily be more secure than authentication using two or more independent means of authentication that happen to be the same factor which are both password combinations (Martin, 2009).

The strength of authentication is dependent on the strength of the authentication factor and mechanism, in addition to the level of identity vetting done at the time of registration. It should not be judged solely by the number of factors involved. In some instances, the authentication protocol may support multifactor authentication, or more than one instance of authentication using the same factor (e.g., entry of a password and a PIN).

#### 4. Form Factors

The forms in which the authenticating factors can be stored and processed include in physical form or memorization by the human claimant. When assessing the form factors for implementation, the assurance level and mobility requirements of the authentication factors to be attained will need to be considered. The most frequently employed physical form factors can be classified into three types, namely; smart card, mobile device and security fob.

Authentication protocols do not dictate the form factors, but simply addresses *how* the secret may be stored, or what kind of access/activation requirements may be in place. That is, the form factor typically does not affect the underlying mechanics of the PoP protocol.

### C. AUTHENTICATION EXCHANGE MECHANISM

The authentication exchange is a key process of the e-authentication protocol. It provides the means of communication between authenticating parties and facilitates the necessary data exchanges for conduct of authentication based

upon PoP of a secret. The authentication exchange mechanism dictates the required transactions and expected data to be sent within the authentication process to establish the success or failure of authentication.

An authentication protocol is often also a cryptographic protocol, as it requires the data exchanged between the authenticating parties to be secure against disclosure of any secrets and any subsequent impersonation attacks. Cryptographic methods deployed in authentication protocols include digital signatures, hashes, encryption/decryption, and challenge-response mechanisms, to name a few.

### 1. Challenge-Response

Challenge-response is one of the most commonly used authentication exchange mechanisms whereby the Verifier will send a challenge to the Claimant, and the Claimant is expected to provide a valid answer in response in order to be authenticated. The simplest form of challenge-response is when the challenge is asks for a password, and the valid response is to provide the correct password directly. It is also used as a form of assertion other than verifying knowledge of a secret. A CAPTCHA (Completely Automated Public Turing Test to tell Computers and Humans Apart), for instance, employs a distorted image of some text which is sent as a challenge, and the valid response is the correct text. A CAPTCHA is used to determine if the authenticated party is a real human rather than a computer program.

In e-authentication protocols, the challenge-response mechanism is typically implemented with one of the cryptographic methods in order to protect the authenticating secret from a direct observation attack. Generally, cryptographic methods can be classified into three categories: symmetric key cryptography, asymmetric key cryptography and hash functions (Kessler, ,2009).

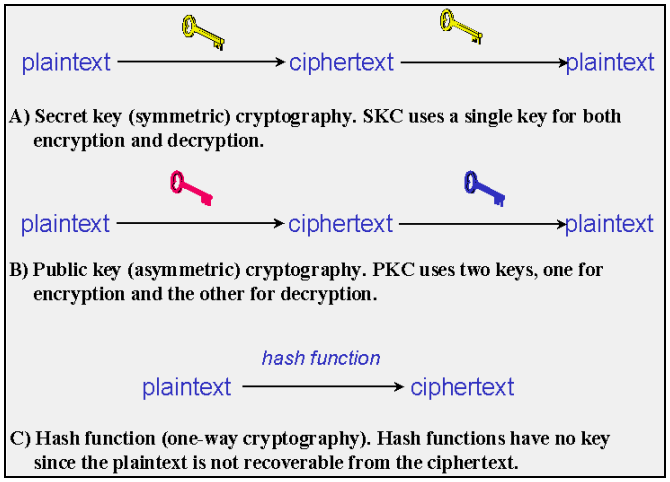


Figure 3. Three Types of Cryptography (From Kessler, 2009)

Symmetric key cryptography is the most commonly seen in challenge-response mechanisms. The challenge-response data exchanges are encrypted with a shared secret key known by both parties to prevent eavesdropping. The figure below shows the execution of challenge response exchanges using a symmetric key. Examples of symmetric key cryptography are AES (Advanced Encryption Standard), DES (Data Encryption Standard) and Triple DES.

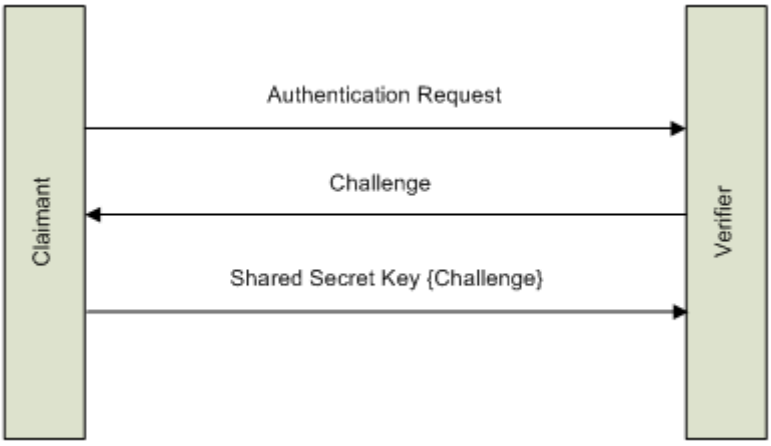


Figure 4. Challenge-Response Using Symmetric Key

Challenge-response mechanisms can also be implemented with asymmetric keys. The Verifier encrypts the challenge using the Claimant's public key. The authentic Claimant is able to decrypt and obtain the challenge using his private key as shown in Figure 5. In a similar fashion, the Verifier may send the challenge in the clear (unencrypted), and the Claimant responds by "signing" (encrypting with his private key) the challenge.

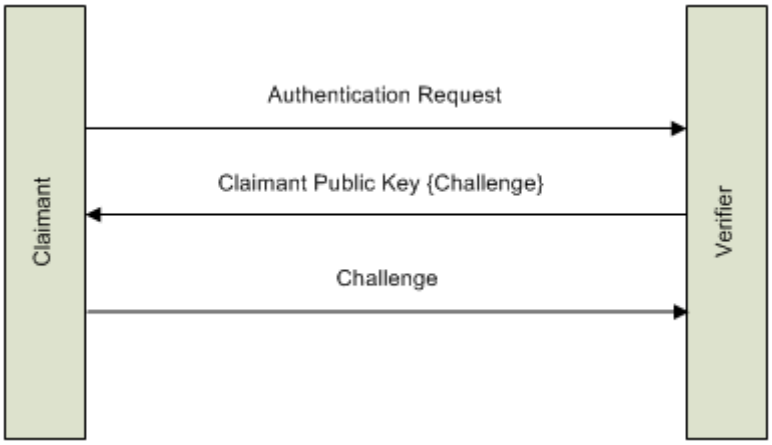


Figure 5. Challenge-Response Using Asymmetric Key

The third alternative is to perform a hash operation on the secret and random challenge to create a valid response. In general, other attributes that need to be exchanged during the authentication process may be hashed to protect the integrity of the data sent across the network.


Figure 6. Challenge-Response Using Hash Function

## 2. Zero Knowledge Proof

Authentication exchange mechanisms that implement zero knowledge proofs are also able to prove possession of secrets without the need to previously convey the secret to the Verifier. Hence, the secret is required to be stored at the Claimant's end only. This maintains the privacy of the secret. However, in order for the Verifier to perform verification that the Claimant indeed possessed the secret, a password verifier will need to be provided by the Claimant to the Verifier beforehand. The password verifier can be generated based on mathematical computation.

The zero knowledge proof is based on the properties of completeness and soundness. The completeness property refers to the Claimant following the protocol closely in order for the Verifier to be convinced by the Claimant. The soundness property refers to the Claimant's ability to prove to the Verifier based on a high probability of success. In the well-known simplistic example, Victor (Verifier) needs to be convinced that Carol (Claimant) knows the secret password to the door connecting paths A and B. If Carol is able to enter from the path A and exit from path B, Victor will be convinced that Carol indeed knows the secret password. The completeness property is illustrated in the example of a protocol that requires Victor to specify the entrance path for Carol to start with, and Carol is able follow the protocol and unlock the door with the secret password without revealing it to Victor. The soundness property is shown in the sense that Carol really knows the secret password if she is able to do this repeatedly, eliminating the probability that Carol may be cheating.

In e-authentication protocols, the zero knowledge proof implementation relies on some mathematical computational model. The protocols are based on hard mathematical problems such as computing the discrete logarithm of large numbers, factorization of numbers, computing the product of large prime numbers, etc.

## 3. Out-of-Band Authentication

Out-of-band authentication uses two separate networks simultaneously to authenticate a user. It allows the use of less secure methods of communicating with the user, and prevents impersonation where the attacker will not only need secret credentials to access the *first* network, but also require secret credentials to access the second (out-of-band with the first) network (Authentify Technology, 2009).

This dual authentication mechanism applies mainly to online transactions, in particular banking transactions (Imperial College London, n.d.). This is implemented by generating a telephone call (the in-band network) followed by an e-mail, or a text message (the out-of-band network) to the user. A typical example is where the user inputs his secret credentials for online authentication, and subsequently receives a Short Message Service (SMS) message on his mobile phone. He will then be required to input the one-time password sent via the SMS as a form of authentication confirmation to complete his authentication process for an online banking transaction.

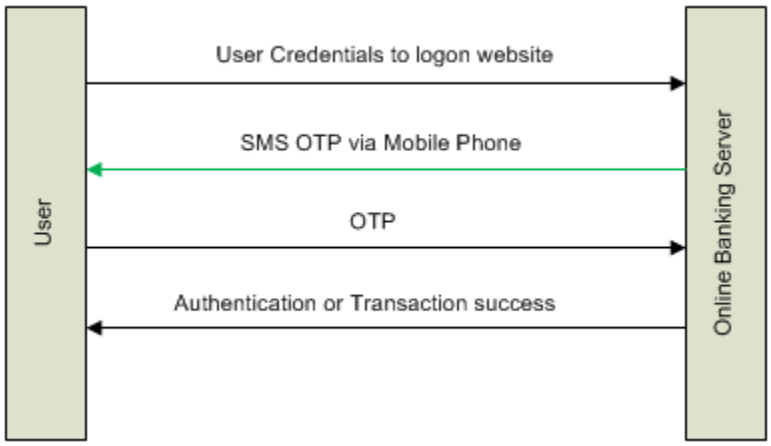


Figure 7. Example of Out-of-band Authentication

### 4. One-way vs. Mutual Authentication

Authentication can take place in a one-way manner whereby only one party (the RP/Verifier) authenticates the other (the Claimant). In mutual authentication, each party has a "stake" in authenticating the other, and thus both parties are RP/Verifier and RP. In the general Web server instance, only one-way authentication is performed (the client authenticates the server as a precursor to sending personal/financial information). Mutual authentication is performed with the authentication exchange mechanism done twice, i.e. doing the one-way authentication in both directions. It is more secure for authentication to be done both ways such that both the client and server are assured of the authenticity of the other party.

## D. PROTECTION AGAINST THREATS

In order for e-authentication protocols to be effective in facilitating communication and data exchanges between authenticating parties, they need to address and protect against the authentication threats. Most e-authentication protocols perform verification and validation to protect against authentication threats such as non-repudiation and impersonation attacks.

Cryptographic techniques are used to protect the integrity and confidentiality of the data being exchanged, and to provide non-repudiation. Symmetric key cryptography can provide confidentiality, authentication and data integrity but *not* non-repudiation. In comparison, asymmetric key cryptography is able to provide these three security objectives *as well as* non-repudiation. There are also other supporting elements such as random number generation, nonces[2], and timestamps that play a significant role within the authentication process to prevent impersonation attacks.

[2] A nonce is a pseudo random number used in authentication protocol to protect against replay attacks and is seldom reused. In some protocols, each client may have a unique sequence number to generate the nonce.

### 1. Impersonation Protection

An impersonation attack is the most fundamental threat to address in authentication. The whole purpose of authentication is to verify identity and ensure that the person is who he claims to be. Well-designed e-authentication protocol exchanges that result in a Claimant proving possession of pre-registered secrets serve to validate the Claimant's "claim" of an identity, and thus prevent impersonation attacks.

### 2. Replay Protection

In the authentication perspective, replay attacks are whereby an adversary can replay valid, previously captured, authentication data in a subsequent authentication session initiated by the impersonator. Replay attacks can be prevented using hash functions that together with a nonce or a timestamp, or both. Hash functions are one-way, irreversible functions that support the ability to compare data contents to identify any modifications, whether accidental or intentional.

### 3. Non-Repudiation

Non-repudiation is an information security objective that is intended to preclude any party that participates in an online/electronic transaction from being able to deny such participation. It is best implemented with digital signature that employ a hash function, a trusted timestamp, and asymmetric key cryptography. Digital signing is performed using the sender’s private key. By signing the data (i.e., encrypting the hash of the data) to be sent along with a timestamp, the recipient can verify and confirm the sender of the data by successfully decrypting the signed data using the sender's certified public key. By keeping the data and corresponding signature on file, the recipient is able to prove not only that the sender did in fact send the data, but also the time at which it was sent.

# III. AUTHENTICATION PROTOCOLS – KEY PLAYERS

## A. OVERVIEW OF CURRENT AUTHENTICATION PROTOCOLS

An authentication protocol entails a sequence of messages exchanged between two parties that allow the use/possession of some secret to be confirmed (Clark and Jacob, 1997). An authentication protocol is also defined as a type of cryptographic protocol with the purpose of authenticating parties wishing to communicate securely (Authentication Protocol, 2009). It is almost certain that any authentication protocol will be dependent on parameters such as names and identities of the authenticating parties, and any secrets shared between them. It is common for public-private key pairs to be used for initial authentication, followed by the establishment and/or transfer of a shared symmetric secret that will be used for the remainder of the session to provide for the integrity and confidentiality of all communicated data.

There are several authentication protocols and mechanisms available. The authentication protocols, just to name a few, include SSL/TLS, Kerberos, EAP, CHAP, RADIUS, IPSec, etc. Each of these authentication protocols employs some common mechanisms for performing authentication, though the implementation may differ in terms of strength and processes involved. Almost all authentication protocols have the feature of using either pre-shared or derived secrets to conduct the identity authentication process. They usually leverage such cryptographic entities as random number generation, hash functions, challenges, nonces and timestamps to enhance the strength or add functionality to the protocol. As a further protocol comparison, some protocols may be *stateful* in facilitating authenticated session resumption, while others may be *stateless* and require periodic re-authentication.

Authentication protocols can be categorized from an application perspective or from the perspective of providing user access to the network and infrastructure. There is typically a separate authentication process for

authentication at the datalink or network layers, compared with the application layer. Authentication for access to the network and infrastructure does not necessarily provide access to applications and services (Todorov, 2007). However, the basic idea of authentication via PoP of secrets applies and remains the same for all categories of authentication protocols, regardless of the layer at which the authentication mechanism is implemented.

The authentication protocol key players surveyed here are a sample of the various categories of authentication protocols available. These key players span from standard protocols for applications and network access, to specific operating system protocols, to independent authentication protocols and to proprietary protocols. The focus will be on the underlying authentication *mechanism*, and will assume that the authenticating parties have already established all required prior configuration, certificate issuances, shared secret key agreement or key distribution requirements.

## B. PAP

The Password Authentication Protocol (PAP) is a simple authentication protocol used to authenticate a user to a network access server. In general, almost all network operating system remote servers support PAP, but it is seldom used. This is due to its insecure nature, as the passwords transmitted over the network for authentication are unencrypted and thus offer no protection against impersonation attacks. The client just sends his user name and password, and the server will send an authentication acknowledgement after verifying the credentials.

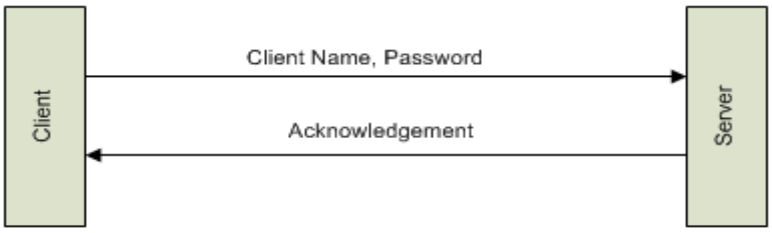


Figure 8. PAP Simple Authentication Message Transaction

## C. IPSEC-ISAKMP

IPSec is considered a meta protocol, which outlines a framework for use of other constituent/component protocols that are developed to provide functionality in specific areas such as authentication, cryptographic key exchange, or key management.

The Internet Security Association Key Management Protocol (ISAKMP) combines the security concepts of authentication, key management and security associations to establish secure communications in an Internet environment. ISAKMP defines procedures and packet formats for key exchange to establish, negotiate, and manage security associations. A security association (SA) describes the security services that are to be established and utilized between two or more parties. SA attributes include items such as the identity of the authenticated party, authentication mechanism, cryptographic algorithm, key length, sequence number, and so on.

ISAKMP is distinct from key exchange protocols; this is to separate the details of security association management and key management from the details of key exchange. It is a framework *for* protocols rather than a protocol itself, as the defined formats provide a consistent framework in key exchange and authentication data, but may utilize several of a number of protocols within this framework. It is independent of any specific key exchange protocol, encryption algorithm, or authentication mechanism. This provides for extensibility in supporting future (or patched/upgraded) algorithms when they may become available, and provides for flexibility in being able to support many combinations of mechanisms and algorithms to fulfill the needs of any specific SA.

A simplified illustration of the ISAKMP framework building block is shown in Figure 9. ISAKMP has basic requirements for authentication mechanisms such as strong authentication and digital signatures. It does not however dictate

any specific authentication protocol within the authentication component. The authentication mechanisms to prove possession of a secret can be via symmetric or asymmetric means.

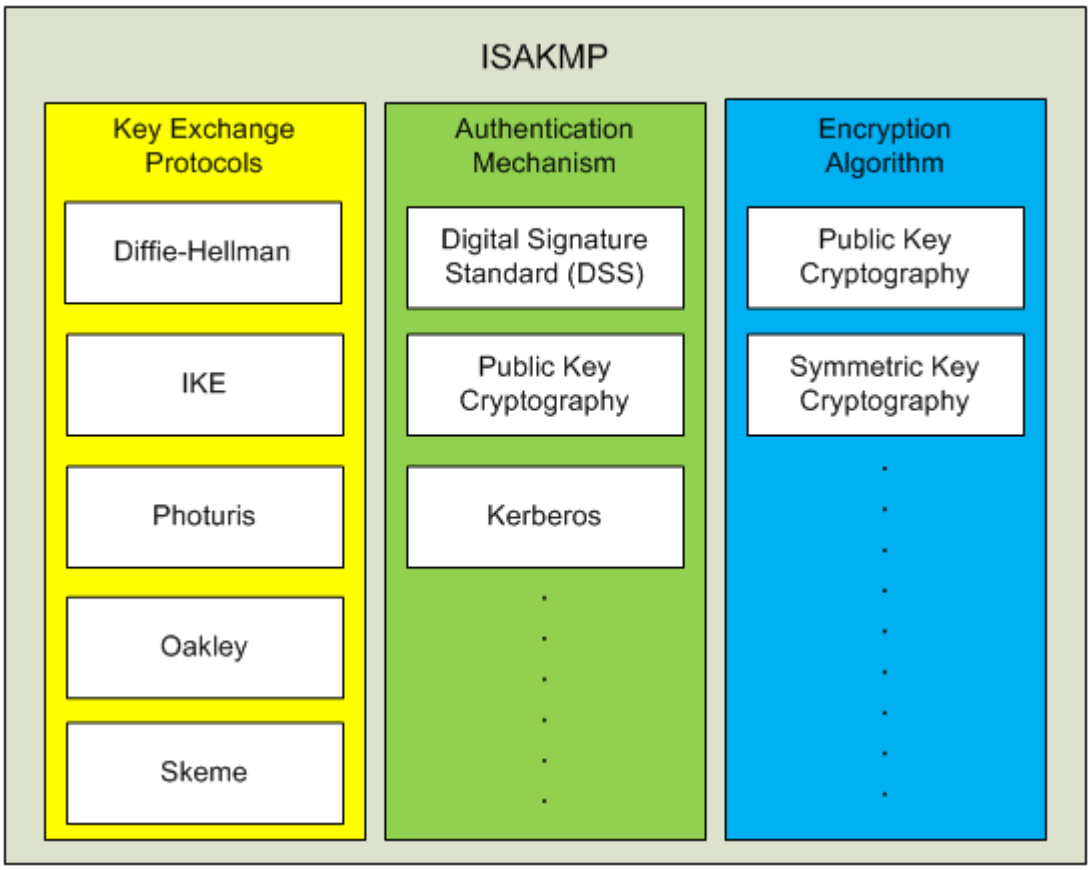


Figure 9. ISAKMP Framework

There are five default message exchanges defined under ISAKMP, namely: Base Exchange, Identity Protection Exchange, Authentication Only Exchange, Aggressive Exchange and Information Exchange (Maughan, et al., 1998). The Base Exchange allows key exchange and authentication information to be transmitted together. The Identity Protection Exchange separates key exchange and authentication information with identity protection. The key exchange is performed with a nonce to protect against replay attacks. The Authentication Only Exchange is used for mutual authentication without key exchange. The Aggressive Exchange minimizes message exchanges by

allowing security association, key exchange and authentication information to be transmitted together without identity protection. The Informational Exchange is used for one-way transmission of information, mainly used for security association management.

## D. IPSEC-IKE

Internet Key Exchange (IKE) is used for performing mutual authentication using some long-term secret key (symmetric secret key, public signature key, or public encryption key) and creates an SA by establishing shared secret keys. An SA is considered unidirectional. For a communication session between two parties, the session will consist of two SAs, one in each direction (Kaufman, Perlman and Speciner, 2002).

IKE defines two phases; the primary objective of Phase 1 is to achieve mutual authentication and establish session keys between the two authenticating parties. Phase 2 leverages the established session keys to facilitate multiple security associations and multiple connections with varying security properties.

In Phase 1, the exchanges and key establishment can occur in one of two modes. Both modes leverage Diffie-Hellman key exchange to establish a session key. Diffie-Hellman is a cryptographic protocol that allows two parties to establish a shared secret key by exchanging messages over an unsecure channel. *Aggressive* mode performs the cryptographic key selection and authentication between Claimant and Verifier in three messages. The proof of identity by the Verifier and Claimant consists of some hash of the pre-shared secret key associated with their identity, the Diffie-Hellman values and nonces. This establishes the Diffie-Hellman session key and verifies that both parties know the shared secret.

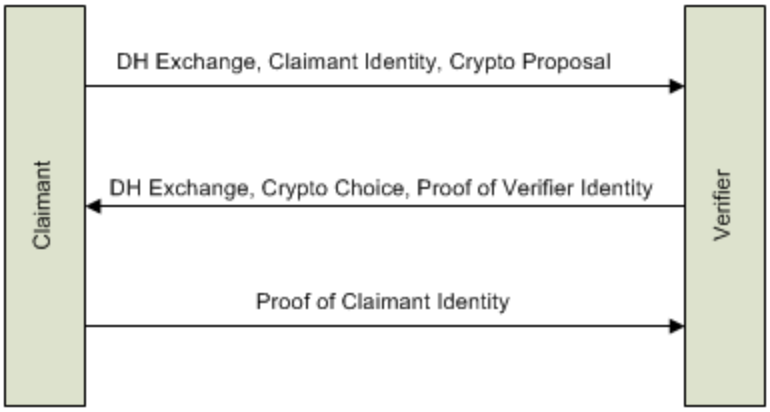


Figure 10. IKE Phase 1 – Aggressive Mode

*Main* mode does the same thing in six messages, whereby the Claimant may propose the cryptographic methods (encryption algorithm, hash algorithm, etc) supported, and the Verifier responds with its choice. IKE in Phase 1 establishes two session keys, an integrity key and an encryption key. With Phase 1 completed, the mutual authentication process is completed, and the IKE security association is set up between Claimant and Verifier.

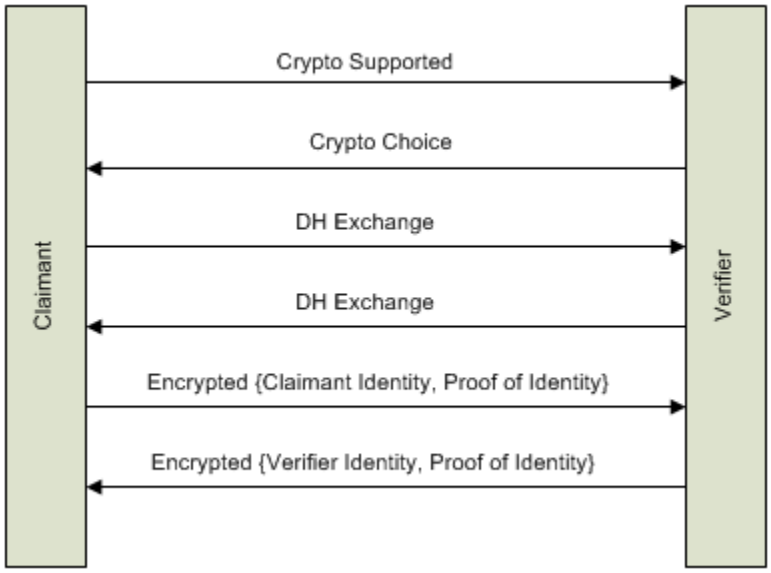


Figure 11. IKE Phase 1 – Main Mode

For Phase 2, there is only one mode known as the *quick* mode exchange. It is used to negotiate protection keys for IPSec. Either the Claimant or Verifier can initiate an IPSec security association. The IPSec security association is considered unidirectional and consists of the cryptographic key, identity of the other end, sequence number, cryptographic algorithm used, etc. This phase involves negotiating crypto parameters and an identifying security parameter index (SPI). The SPI is used to uniquely identify the security association.

Diffie-Hellman exchange is optional in this phase. Phase 2 protocol exchanges are accomplished in three messages. Phase 2 exchanges include the sending nonce and other information which; together with the key material seed computed in the IKE Phase 1, are used to compute and generate the integrity and encryption keys for the IPSec security association. All messages in Phase 2 are encrypted with the encryption key established during Phase 1.

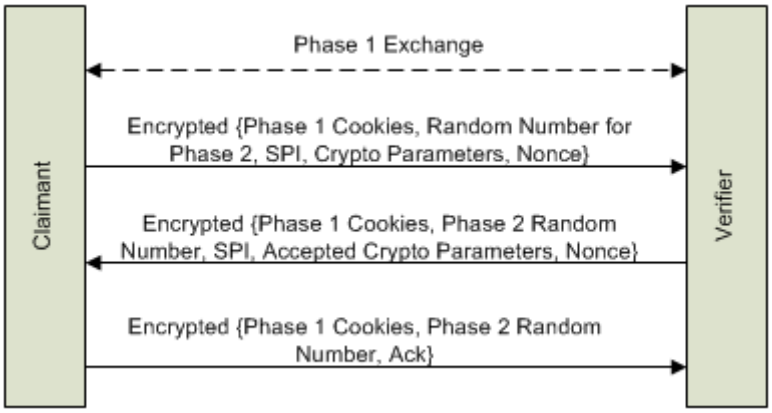


Figure 12. IKE Phase 2 – Quick Mode

### 1. Photuris

Photuris is a session key management protocol for IPSec and was one of the potential candidates for IKE. The protocol establishes session keys between two communicating parties without the need to exchange the session keys over the communicating medium. The authentication mechanism is based on a

shared secret key between the communicating parties using Diffie-Hellman and cookies. Photuris' use of cookies was designed to provide some form of protection for denial of service attacks. A cookie can be a chosen random number sent by the party who initiated the communication or the receiving party to ensure the source IP address of the initiator and to track the communication connection. The protocol also provided for the cookie to be stateless. The cookie can be the output of a hash function that uses the IP addresses of communicating parties and a secret known only to the cookie owner (Kaufman, Perlman and Speciner, 2002). In this case the cookie owner does not need to "remember" the cookies that have been sent; the cookies can simply be computed on-the-fly based on the destination party's IP address.

The initial message exchanges include cookies, crypto negotiation, and Diffie-Hellman data used to establish the session key. Thereafter, the session key is used to encrypt any other security parameters exchanged between the two parties. The session keys generated by the Photuris protocol are meant to be short-lived since the secrets, known only to the cookie owner, may be reused for several communication connections. The session key may be changed periodically through additional message exchanges (P. Karn, 1999).

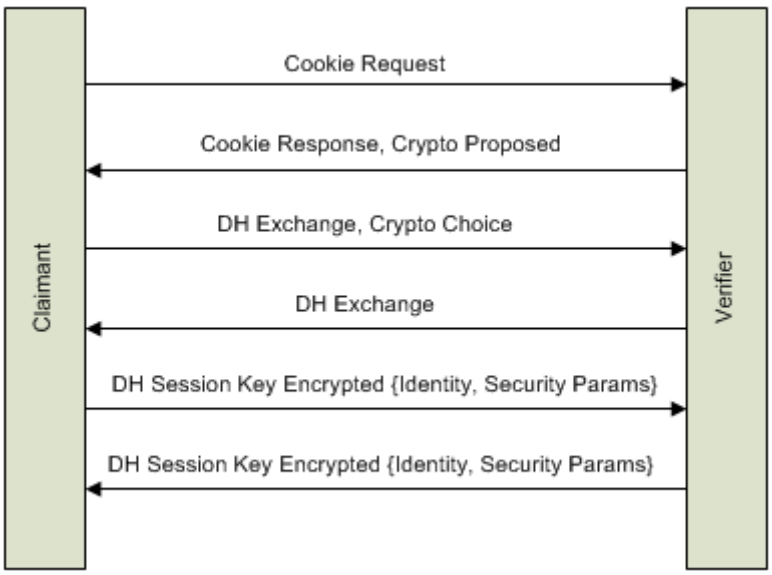


Figure 13. Photuris – Simplified protocol exchanges

### 2. Oakley

Oakley is a key determination protocol that is designed to work within the ISAKMP framework. The protocol aims to establish a secret key between authenticated parties that can serve for a long lifespan. The three key components of the protocol are the cookie exchange, Diffie-Hellman key exchange and authentication (Orman, 1998). This is similar to Photuris in using Diffie-Hellman for the key exchange mechanism for deriving a shared secret key as well as using cookie exchange in preventing denial of service attacks.

The method of authentication can be via digital signatures, public key encryption, or an out-of-band symmetric key. Both authenticating parties can exchange nonces and a pre-shared secret key to achieve the authentication and derive keying material. PoP of an asymmetric secret is used if non-repudiation is required.

## E. SSL/TLS

Secure Socket Layer (SSL) / Transport Layer Security (TLS) is an authentication protocol that allows two parties to authenticate and establish a shared secret key used for a secure communication session. SSL/TLS is a stateful protocol where the client and server maintain current connection state information. SSL/TLS runs over TCP and is usually used to authenticate and protect data exchanges between client and server.

SSL/TLS begins with an initial handshaking phase between the client and server to negotiate the protocol version, cryptographic algorithm, and a random number. After that is the key exchange and authentication phase, where the server public key certificate is sent to the client for verification and a shared secret session key (master secret key) is created based on a random number selected by both parties.

The authentication is performed as follows: the client challenges the server by encrypting a selected random number (pre-master secret, S) with the

server public key, and by encrypting some keyed hash of messages with the master secret key. The server authenticates to the client by being able to decrypt the pre-master secret with its private key to obtain S, and it subsequently generates the Master secret key (K) which it uses to decrypt the keyed hash of messages. The server responds with the decrypted keyed hash of messages to complete the authentication process. Henceforth, both parties use this shared secret session key to derive keys for encryption and integrity protection in their subsequent secure communication and data exchange. This is typical of most authentication protocols that leverage public key encryption for initial exchange of authentication info and that generate a shared secret key for subsequent communication and data exchanges.

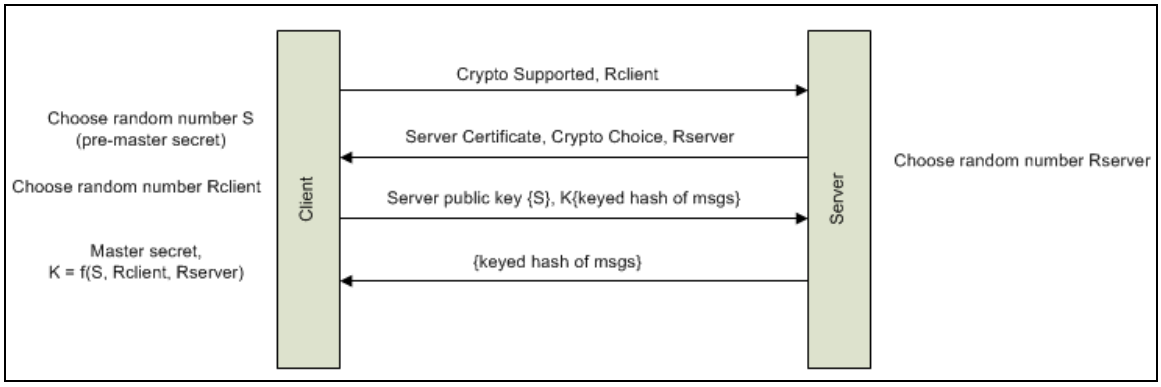


Figure 14. Simplified SSL/TLS typical authentication exchange

SSL/TLS allows session resumption when a session terminates, and the client wants to re-establish the session with the same server using the same connection parameters. The client will initiate the session resumption during the handshake process using the previous session ID. The server will need to maintain some state based on the previous authenticated session and replies with the same session ID if willing to resume the session. Both parties can then perform data exchanges based on previous session shared secret keys, resulting in a shortened handshaking process. If the server does not recognize the session ID, a different session ID is sent to the client followed by the complete

handshaking process (Kaufman, Perlman and Speciner, 2002). The previous session in this case is then non-resumable.

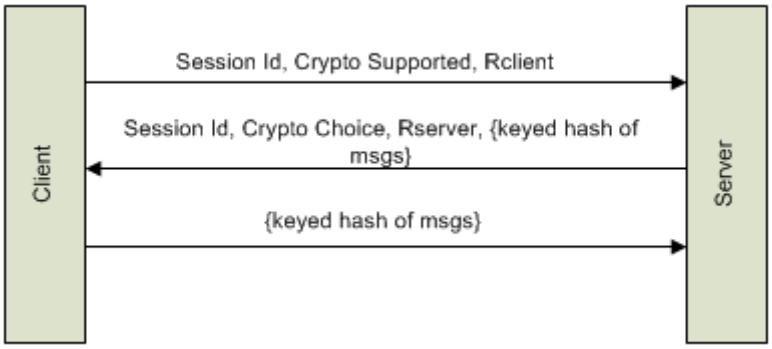


Figure 15. Session Resumption (using previous session ID)

Another key aspect for authentication is that it uses X.509 certificates for peer-to-peer authentication and is able to support both one-way and mutual authentication (Todorov, 2007). In most commercial deployments, it is usually one-way authentication where only the client authenticates the server. Mutual authentication, though supported, is seldom used.

## F. KERBEROS

Kerberos is a secret key-based authentication protocol for networks. A Kerberos implementation consists of a Key Distribution Center (KDC) deployed at a secure physical site in the network. A KDC works as a trusted third party to facilitate parties who wish to authenticate and communicate securely with one another. When Alice wishes to talk to Bob, she will need to go through the KDC to obtain a session key. The KDC is responsible for generating a shared secret key (master key) for Alice and Bob's secure communication session.

Suppose a client initiates a request to the KDC to establish a communication session with a server. The KDC will generate a shared secret session key for the client and server. The shared secret key and client name are encrypted using the client and server's already established shared secret key (long-term secret password). This is referred to as a Kerberos ticket. Based on

this ticket, the client and server can authenticate each other, and they are able to use the shared session key for subsequent communication.

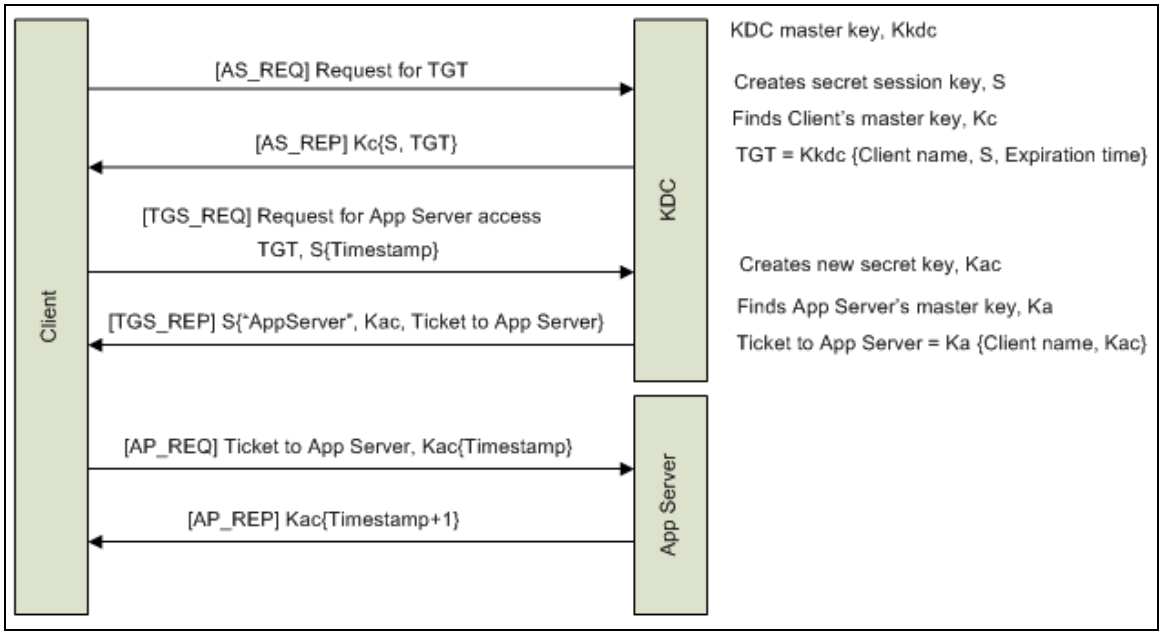


Figure 16. Kerberos Authentication Process

In the request to the KDC for a session key, the KDC sends a ticket granting ticket (TGT) which includes the session key and other information such as the TGT expiration time. This is to restrict the validity time of the session key so that even if it is compromised, it is only for a limited period. This session key and TGT can be used for later service requests to access the services or resources without the need to do re-authentication (Todorov, 2007). The ticket lifetime can be specified to restrict the validity of the ticket. Kerberos prevents replay attacks with the current timestamp on its service tickets. However, there is a need to synchronize time on all involved authenticating parties (KDCs, clients and servers) to support the usage of timestamps.

In accessing and seeking authentication to a server in a remote realm or domain, a client can be authenticated using referral tickets generated by the client local KDC. The client will use this TGT from his own domain and present it to the KDC in the targeted domain. Upon verifying the TGT presented and

determining if the requested server access is within its realm, the remote KDC will issue another TGT to the client. Using this newly issued TGT, the client is able to access the remote server. The inter-realm trust relationships is built in a stateless way by TGT, timestamp on the service ticket and the trusted third party concept based on the deployed KDCs.

## G. SRP

Secure Remote Password Protocol (SRP) is a non-disclosing, secure password-based authentication protocol. The protocol facilitates a client authenticating to a server based on a zero-knowledge proof and without need of a trusted third party. The key feature of this protocol is that there is no need to reveal the actual secret password of the client to anyone. The authentication depends on the password verifier that is generated by the client to be pre-shared with the server. Note that this verifier is a necessary constituent building-block enabling verification of the client's secret, but it is not the secret itself. The server authenticates the client based on this password verifier to verify that the client indeed possesses the secret password. It is considered a non-disclosing authentication protocol and offers complete protection against both passive and active attacks (T. Wu, 2000). SRP is based on a computationally difficult mathematical model and large random number properties to achieve the zero knowledge proof.

In general, the client initiates the authentication protocol. Upon identification to the server, the client will receive the salt[3] stored in the server under her username. The client generates a random number, raises a primitive root modulo the power of the selected random number and sends the result to the server. This is done similarly at the server's end in addition to the password verifier associated with the client. Both sides are then able to construct a shared

[3] Salt refers to random bits or random number used in cryptography or some derivation function to generate secret key.

session key. Thereafter, they need only to prove to each other that their keys match to complete authentication. This enables both parties to communicate securely after successful authentication.

```
 N    Large prime number
 g    Primitive root modulo N
s    Client's salt
P    Client's password
H()  One-way hash function
^    (Modular) Exponentiation
u    Random number
a,b  Secret ephemeral values
A,B  Public ephemeral values
x    Client's private key
v    Password verifier
S    Shared secret session key
```

| | Client | | Server |
|---|---|---|---|
| 1. | | $C$ --> | (lookup $s$, $v$) |
| 2. | $x = H(s, P)$ | <-- $s$ | |
| 3. | $A = g^a$ | $A$ --> | |
| 4. | | <-- $B$, $u$ | $B = v + g^b$ |
| 5. | $S = (B - g^x)^(a + ux)$ | | $S = (A \cdot v^u)^b$ |
| 6. | $K = H(S)$ | | $K = H(S)$ |
| 7. | $M[1] = H(A, B, K)$ | $M[1]$ --> | (verify $M[1]$) |
| 8. | (verify $M[2]$) | <-- $M[2]$ | $M[2] = H(A, M[1], K)$ |

Figure 17. SRP Authentication Process (After T. Wu, 1997)

In this protocol, only the client generates a secret password and computes a corresponding verifier. To establish a password verifier with the server, the client picks a random salt and computes the hash based on the secret password and salt. The password verifier is the result of the computation of the primitive root modulo. The server will have a verifier for each client that allows it to authenticate the respective client. If this verifier is compromised, the attacker will still not be able to impersonate the client due to the one-way function on the secret password to create the verifier (T. Wu, 1997).

The password verifier in this protocol can be seen as a pre-shared secret between the client and the server. It acts as the key authentication factor that allows the server to authenticate the client. From another perspective, such an authentication mechanism is similar to PKI authentication using public-private keys. The secret password in this case is analogous to the private key, and the password verifier is analogous to the public key. In addition, the secret password and password verifier are mathematically related. Similar to the public key distribution concept, the password verifier can be publicly distributed to whomever needs to authenticate the client. Conceptually, asymmetric secret authentication seems applicable to this protocol, taking a different form of implementation.

Unlike authentication using asymmetric secrets, trusted key servers and certificate management infrastructures are not required. It also prevents the need to store the client's secret password at more than one location for authentication purposes. The secret password never leaves the client's local machine, and there is no need to store the secret password at the server's end. Thus, it is protected against password database attacks at the server, preventing a stolen password from being used in an impersonation attack.

## H. TELNET

Telnet authentication is a very simple process that consists of a primitive login where the server requests both username and password, and the user provides both in plaintext. This is similar to the PAP.

Subsequent versions of Telnet support authentication options that provide mechanisms for more secure authentication negotiation between the client and server (Todorov, 2007). This allows the client and server to agree on specific authentication protocols for credential exchange, proof of identity and subsequent data protection. The supported authentication protocols include Kerberos, SRP, SSL, among others.

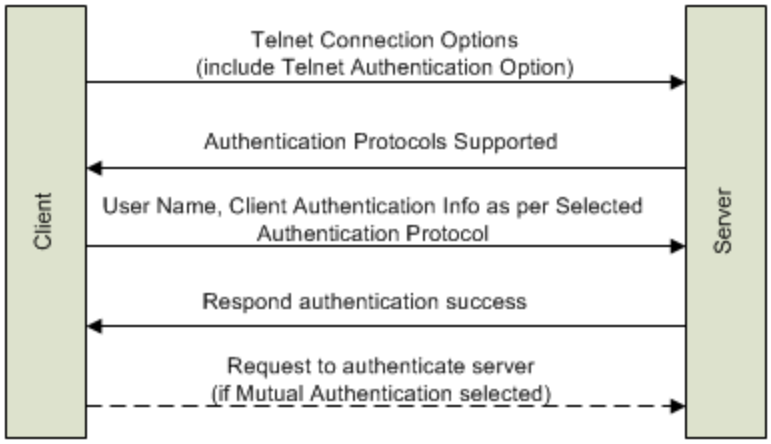


Figure 18. Telnet Authentication – Using Authentication Option

## I. SSH

The Secure Shell (SSH) protocol provides a secure (encrypted and authenticated) channel to facilitate remote login over unsecured networks and allows secure data exchange between two hosts or networked devices. It was designed to facilitate protected data exchanges for applications, such as Telnet, that lack built-in authentication and data protection. Typically, SSH uses public key cryptography to authenticate remote users. The client verifies the identity of the server using the public and private key pair.

The transport layer authentication deals mainly with server authentication with an asymmetric key pair. The two SSH versions are similar in terms of how authentication is done; the only difference is in the way the session key is generated (Todorov, 2007).

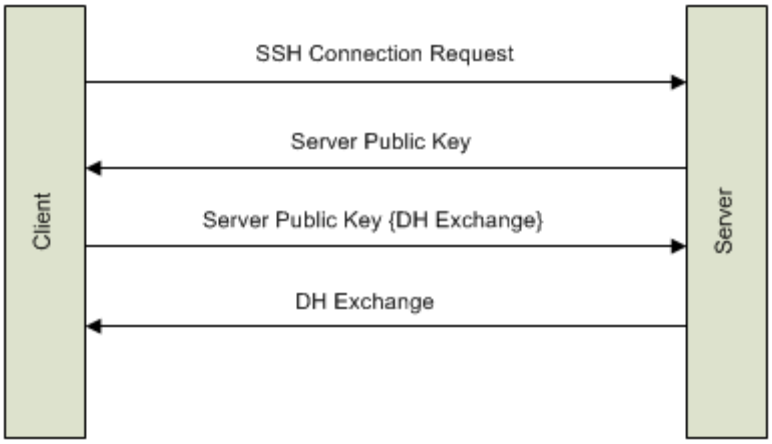


Figure 19. SSH Server Authentication

Once the transport layer negotiation completes, user authentication is performed. The supported authentication methods include Public Key authentication, Host authentication, Password authentication, and none (Ylonen and C. Lonvick, 2006). Public Key authentication is the default authentication mechanism that SSH clients and servers are required to support. Both Public Key and Host authentication leverage an asymmetric key pair for authentication. The client authenticates to the server by encrypting the authentication request message with the client private key. The server verifies and responds with authentication success if the request message is decrypted using the client public key. Host authentication is similar with the authentication request from a remote host signed using the remote host private key. The Password authentication method provides for user authentication using a plaintext username and password. It is considered secure to use plaintext for authentication since SSH is a secure communication channel. The server may also use external—3rd party—authentication protocols (e.g., Kerberos) for the user authentication process.

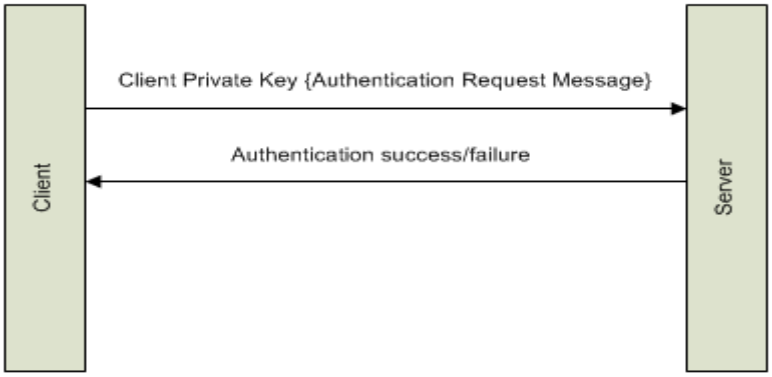


Figure 20. SSH User Authentication – Public Key Authentication

## J. CHAP

The Challenge Handshake Authentication Protocol (CHAP) defines an authentication method, which uses a random challenge with a cryptographically hashed response constructed using the challenge and a secret key (Simpson, 1996). The challenge-response mechanism provides protection against replay attacks through the use of an incrementally changing identifier and a variable challenge value. The authentication method depends upon a shared secret known only to the Claimant and Verifier. This secret is not sent over the communication link. Although the authentication is typically one-way, by negotiating CHAP in both directions the same shared secret is used for mutual authentication.

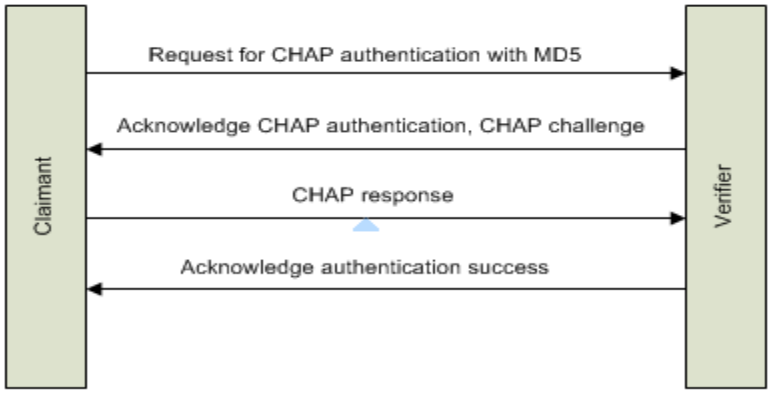


Figure 21. CHAP Authentication Process

Both parties will need access to the shared secret in order to generate the required hash for CHAP authentication. This will mean that the plaintext shared secret will need to be stored by both Claimant and Verifier. This creates a vulnerability to password database attack. Microsoft came up with its own version of CHAP (MS-CHAP) which rather than being a completely separate authentication protocol, simply use a different algorithm for generating the hash (Zorn, 2000).

## K. EAP

The Extensible Authentication Protocol (EAP) is an authentication framework that supports multiple authentication methods. The EAP framework is flexible in allowing the Verifier to determine the specific authentication method to be used rather than supporting one specific authentication mechanism. EAP defines four types of packets, namely *request*, *response*, *success,* and *failure* (Aboba, et al., 2004). The Verifier issues request packets, and a response packet is obtained from the Claimant. The Verifier sends success or failure packets after completion of the authentication procedures. EAP also supports backend authentication servers that implement some or all authentication methods. This serves as pass-through authentication to send to the remote EAP authentication server, which may be using protocol such as RADIUS. This facilitates centralized management of authentication for large numbers of Verifiers.

The three authentication EAP types – MD5 challenge, OTP, and GTC are not considered sufficiently secure for typical uses; this applies in particular to wireless environments (JANET Technical Sheets, 2007). The MD5 challenge is a challenge-response messaging mechanism. OTP is similar but uses a one-time password in the challenge-response mechanism. Generic Token Card (GTC) is for use with token card implementations that require user input.

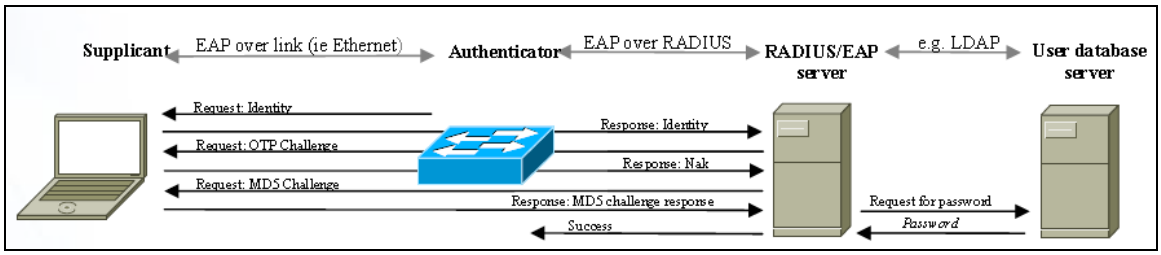


Figure 22. Example of EAP Authentication (From JANET Technical Sheets, 2007)

As with all EAP mechanisms, the initial authentication phase is unencrypted and not protected. Some of the more secure types used are EAP Transport Layer Security (EAP-TLS), Protected EAP (PEAP) and Tunneled TLS (TTLS) (Todorov, 2007). EAP-TLS is an authentication mechanism that uses a user's certificate to authenticate the Claimant to the server/Verifier. EAP-TTLS is an extension that allows authentication using other authentication mechanisms such as PAP or CHAP. PEAP and EAP-TTLS are similar in using TLS for server authentication and encryption. Neither PEAP nor EAP-TTLS require user certificates by using another authentication protocol between the Claimant and server that is protected by TLS encryption.

## L. RADIUS

Remote Authentication Dial-In User Service (RADIUS) is an authentication protocol used in network environments, commonly used for embedded devices such as routers, modem servers, switches, etc. It is a widely accepted de-facto standard for remote authentication and authorization to infrastructure access. A RADIUS server is typically responsible for accepting user connection requests and authenticating users to facilitate delivering service to users after successful authentication (Rigney, et al., 2000).

RADIUS is a stateless protocol utilizing a model of trust based on a shared secret between client and server, and it permits using the same shared secret by many clients. By allowing the use of the same shared secret, this results in a single compromised client, effectively compromising other clients who

share the same secret (Hill, 2001). Typical authentication is via simple authentication using user password or a challenge response mechanism (Todorov, 2007).

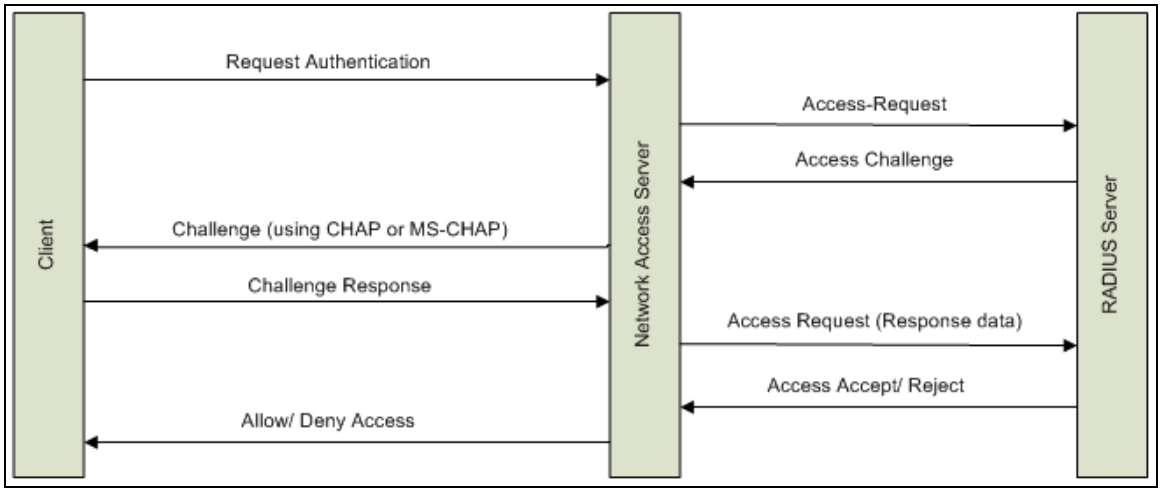


Figure 23. Challenge Response Authentication Using RADIUS

## M. NTLM

NTLM is an authentication protocol by Microsoft and is primarily used by Microsoft operating systems. It requires a secure channel and a persistent TCP connection between trusted parties for client-server authentication. The general NTLM authentication process involves shared secret processing between authenticating parties, a challenge-response mechanism, and the computation of NT and LM (Lan Manager) hash values. NTLM typically uses a derivative of the client password to encrypt a challenge string. The authentication mechanism is able to prevent replay attacks; however, it is vulnerable to a man-in-middle attack.

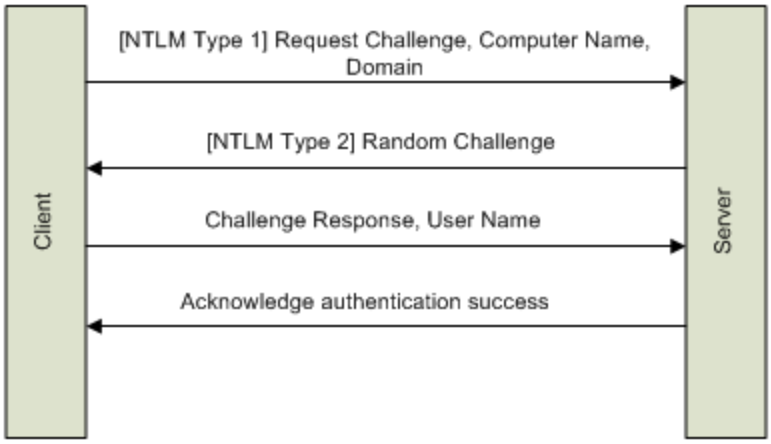


Figure 24. NTML Authentication Process

## N. TACACS+

TACACS+ (Terminal Access Controller Access-Control System Plus) is a protocol that provides access control for routers, network access servers, and other networked computing devices via one or more centralized servers. TACACS+ provides separate authentication, authorization, and accounting (AAA) services.

The key differences between TACACS+ and RADIUS are that TACACS+ separates authentication and authorization operations, whereas RADIUS combines them. Further, TACACS+ uses TCP while RADIUS uses UDP. The similarity is that TACACS+ authentication is also based on a shared secret between an infrastructure devices and the TACACS+ server. The shared secret is used to authenticate both the client and server by encrypting the communication between both parties. The assumption is that the client is considered authenticated if it can successfully decrypt messages sent to it. The same applies to authenticating the server. There is an anti-replay mechanism by means of packet sequence numbers used to calculate the protection key; however, they are easily predictable since the sequence numbers always start from 1 (Todorov, 2007).

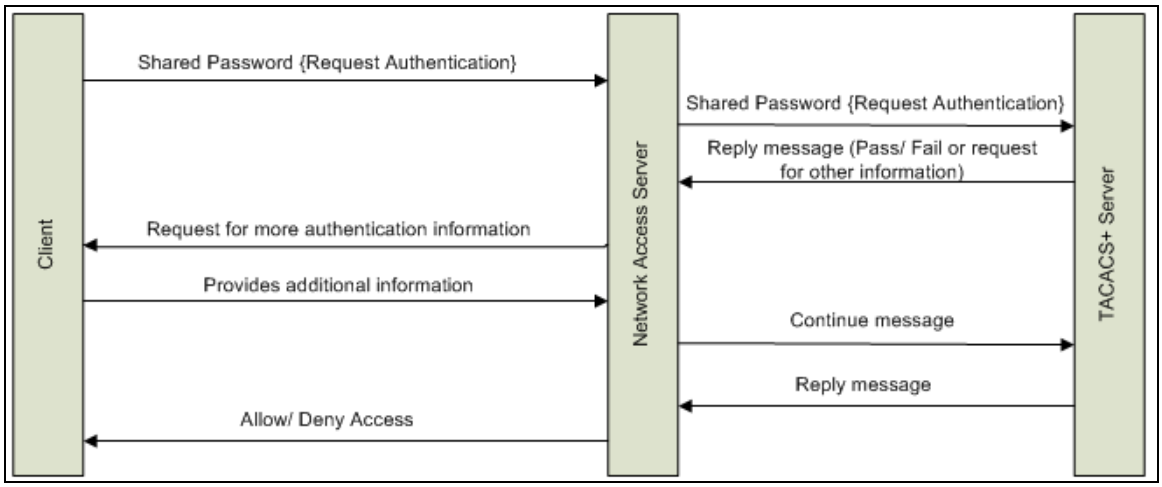


Figure 25. TACACS+ Authentication Process

## O. WIRELESS AUTHENTICATION

Under the 802.11 standard, the two main authentication methods to access the Wi-Fi infrastructure are open authentication and shared key authentication. The open authentication method is almost equivalent to no authentication at all. No verification of the client's requesting a connection is required for a Wi-Fi access point to grant connection access. However, if Wired Equivalent Privacy (WEP) or Wi-Fi Protected Access (WPA) is configured for the network, the client will not be able to connect to the infrastructure unless the encryption key for WEP or WPA is known. WEP and WPA are enabled to protect the data traffic within the wireless network. The data protection algorithm mainly consists of deriving a common secret key between the client and the access point (Todorov, 2007).

The shared key authentication is based on a challenge-response mechanism. The access point sends a random challenge to the client for authentication. The client uses WEP to encrypt the received challenge and returns it to the access point. However, WEP resulted in an attacker's ability to recover and decrypt other parties' encrypted message even without knowing the encryption key used between the two parties (Goldberg, 2001).

The WPA/WPA2 pre-shared key method is used instead. It is similar to the WEP concept in that the client and the access point share a common secret key. The secret key is used to authenticate the client as well as for protecting user data in terms of data traffic encryption and ensuring data integrity.

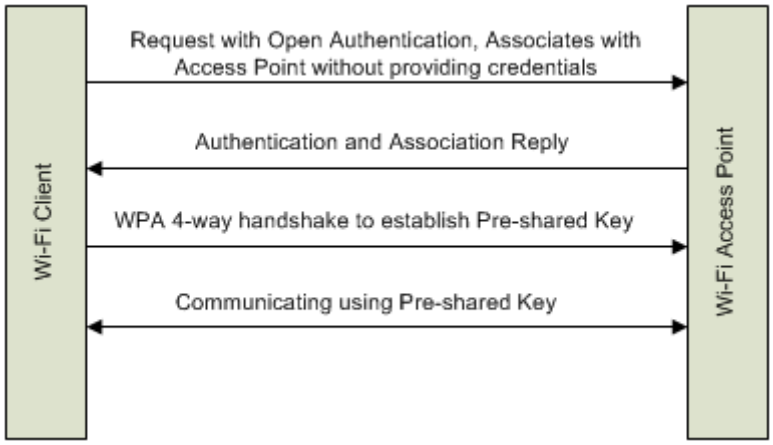


Figure 26. WPA Pre-shared Key Authentication Process

## P. VPN AUTHENTICATION

A Virtual Private Network (VPN) connects two remotely located endpoints together across a public network. The authentication protocols used for VPN are typically, PAP, CHAP, MS-CHAP, EAP and SSL (Lancaster, 2002).

In general, authentication takes place primarily at the transport and application level. First, transport level authentication is performed through the exchange of computer certificates or a pre-shared key (IPSec-IKE) during the establishment of the IPSec security association. This is followed by application level authentication, in which the remote access client that requests the VPN connection is authenticated through the use of a Point-to-Point Protocol (PPP) authentication method (EAP, CHAP, MS-CHAP). These authentication protocols are covered in the earlier sections of this chapter. Upon successful authentication, a secured VPN tunnel is established between the client and the server.

## Q. GSM AUTHENTICATION

The Global System for Mobile Communication (GSM) is a standard for digital cellular services. Authentication and encryption are integrated into GSM through the Subscriber Identity Module (SIM) card and serve to identify the subscriber. The SIM card includes subscriber information and the International Mobile Subscriber Identity (IMSI), which is a unique 15-digit code, used to identify an individual user on a GSM network (GSM Security FAQ, 2003).

A3 is the authentication algorithm used in GSM systems. It is secret key based and authenticates using a challenge response mechanism. The A3 authentication algorithm takes the random challenge received by the SIM as one of its inputs. The other input is the secret key residing in the SIM. From these two inputs, the A3 algorithm generates the secret response. COMP128 is the default algorithm implementation for the A3 algorithm used by GSM network operators for authentication and key exchange (Thakker, n.d.) (Chen, 2002).

GSM authentication is based upon symmetric keys that have been pre-loaded onto each subscriber's SIM chip. All subscribers' keys are also stored in several HLRs (home location registers) that are under the control of the phone network. The A3 algorithm runs on the SIM, computing cryptographic authentication responses when necessary. Only legitimate SIM chips can provide the correct response to the random challenge. Also, the IMSI of every SIM is unique, and no two SIM cards can have the same IMSI.

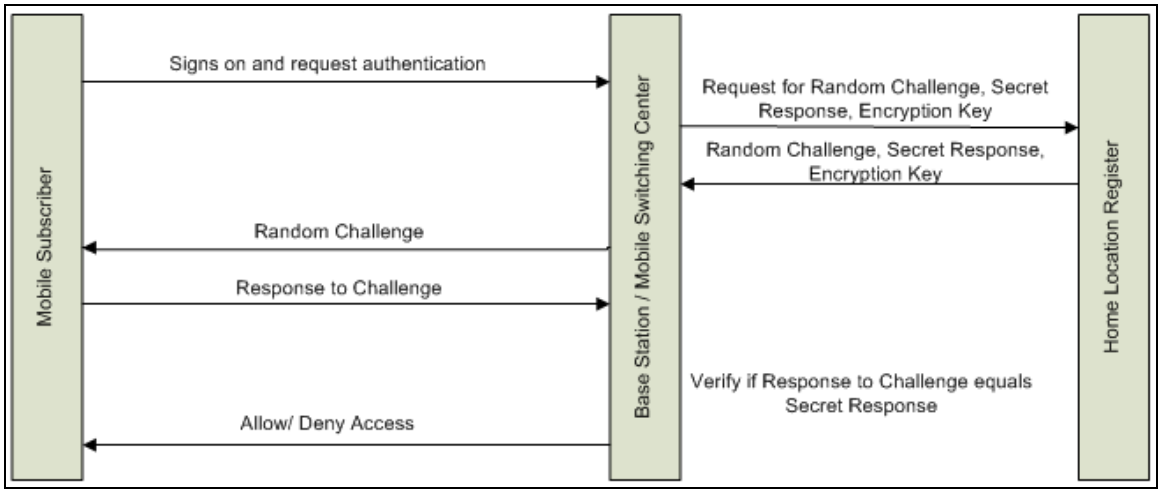


Figure 27. GSM Authentication Process

## R. E-VOTING AUTHENTICATION PROTOCOL

Performing e-voting transactions over the internet is being continually refined, and in particular, the authentication protocol to support this type of transaction. An e-voting authentication protocol requires the classic information security properties of protection against impersonation, observation, and modification attacks. Additionally, a unique property required for an e-voting authentication protocol is deniability. A deniable authentication protocol allows the receiver to authenticate the sender of a message in a way that the receiver is unable to prove the source of the message to a third party. This is critical to ensure the privacy of a vote.

There are several proposed deniable authentication protocols, which are based on zero-knowledge proofs, factoring, or a discrete logarithm. The following shows an example of a proposed deniable authentication protocol based on a discrete logarithm problem (Meng, 2009).

During the initialization phase, the Authority is required to choose a large prime number as well as to compute a random number. Based on the large prime number and random number posted by the Authority, as well as a random

number picked by the sender, the sender computes his public and private keys. The sender may compute a series of public keys based on a series of random numbers and post these public keys publicly.

During protocol execution, the sender randomly picks a public and private key from his series of key pairs generated previously. A hash is computed based on his private key and vote message. The Message Authentication Code (MAC) is then computed based on the receiver's public key, the hash from the previous computation, and the vote message. Then, the sender public key, MAC and vote message are sent to the receiver. The sender "forgets" the used private key after a certain time. The MAC serves to ensure integrity of the vote message, with the sender public key indicating which key pair the sender used in this transaction.

The sender's ability to deny having ever authenticated anything to the receiver is based on the multiple key pairs generated. Also, the sender will be unable to provide his private key since the sender "forgets" it after each transaction. The authentication mechanism in this protocol is based on asymmetric key authentication with a variant from the typical public-private key authentication in that there is more than one key pair generated per user, and the asymmetric keys are short-lived.

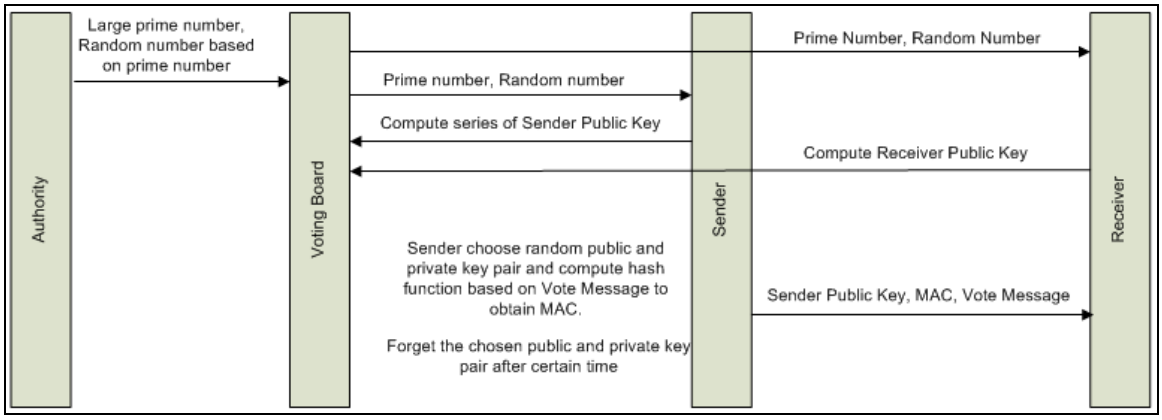


Figure 28. Example of E-Voting Protocol Authentication

## S. MIFARE PROPRIETARY PROTOCOL

Mifare Classic is a commonly used contactless smart card, used mainly for payment in public transportation systems. Contactless cards are based on Radio Frequency Identification Technology (RFID). The Mifare Classic RFID chips use a mutual authentication process to authenticate both the card and reader. The proprietary protocol design and implementation details are kept secret.

Through a study and experiments conducted to analyze the communication between the card and the reader, it was discovered that the Mifare Classic uses symmetric keys and that there exists a weaknesses in its psuedo-random generator (Gans, Hoepman and Garcia, 2008). This weakness enables the recovery of keystreams (i.e., temporary keys derived from long term keys) without knowing the long-term encryption key. It was discovered that the authentication protocol performs a four-step mutual authentication between the card and reader that can be subjected to a replay attack. A trace of a successful authentication can be replayed multiple times until a challenge nonce equal to one processed in the original (recorded) trace is provided by the authenticator.

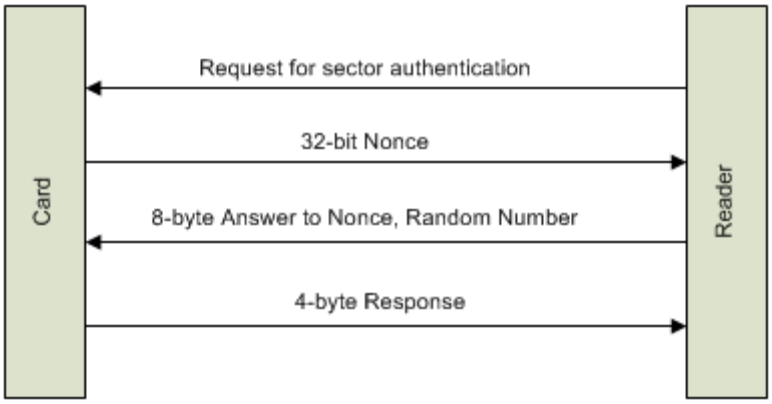


Figure 29. Mifare Classic Authentication Process

# IV. BUILDING AN AUTHENTICATION PROTOCOL TAXONOMY

## A. NEED FOR A TAXONOMY

There are several e-authentication protocols available. Each protocol is designed for a specific application and operating environment to ascertain the identity of the requestor and to thus protect access to resources and data. Depending on the e-authentication protocol mechanisms implemented, the protocol can achieve a certain security assurance level in providing reliable identity verification. The authentication protocols also incorporate mechanisms to address common authentication threats such as repudiation, impersonation, modification, and observation attacks.

The taxonomy for the e-authentication protocols is to facilitate understanding of the protocol characteristics and the intrinsic mechanisms involved in the authentication operation. The classification enables one to distinguish the similarities and differences among these authentication protocols and to provide some basis for protocol evaluation and/or selection. This can be leveraged to assess and select potential protocols for specific system requirements and problem domain deployment.

## B. CLASSIFICATION CRITERIA

There are many ways that e-authentication protocols can be classified. The basis for classification is dependent on the application of the developed taxonomy. With the objective of using this developed taxonomy for the selection of potential authentication protocols to meet specific system requirements or problem domains, the proposed classification is based on performing a functional decomposition of the protocol.

The focus is placed on examining the authentication transactions that are required between the authenticating parties. Functional decomposition serves to

identify the functions and mechanisms of the protocol in facilitating the authentication process. In addition, key data elements are identified from the data exchanges that take place during the authentication session. This includes how the PoP of a secret is carried out and verified, as well as how authentication threats, if any, are addressed.

The key functions, mechanisms and critical components of the authentication protocol are broken down into the classification criteria as described below.

- *Authentication Factor* defines the key component, which is the secret in an authentication session.
- *Secret Protection* defines how the secret is to be protected throughout the authentication session.
- *Authentication Methods* defines the various authentication mechanisms employed.
- *Support Elements* recognizes any additional data elements that are transacted in an authentication session, and what characteristics they contribute.

The proposed taxonomy is composed of all the classification criteria set herein.

## C. THE PROPOSED TAXONOMY

The proposed taxonomy for e-authentication protocols is based on authentication factor, secret protection, authentication methods, and support elements. To use an analogy, the e-authentication protocol is the *language* to be used for authentication. The authentication transactions may be seen as sentences that are required to convey discussion elements between authenticating parties. The sentence structure is dictated by the authentication methods that provide the *semantics*. The authentication factor, which refers to

the secret to be proven and verified, can be seen as the *subject* of the sentence. The secret protection represents the communication *medium* to protect the communication of the secret. Finally, the key data elements under the support elements category represent the *adverbs and adjectives* that are transacted in support of the secret.

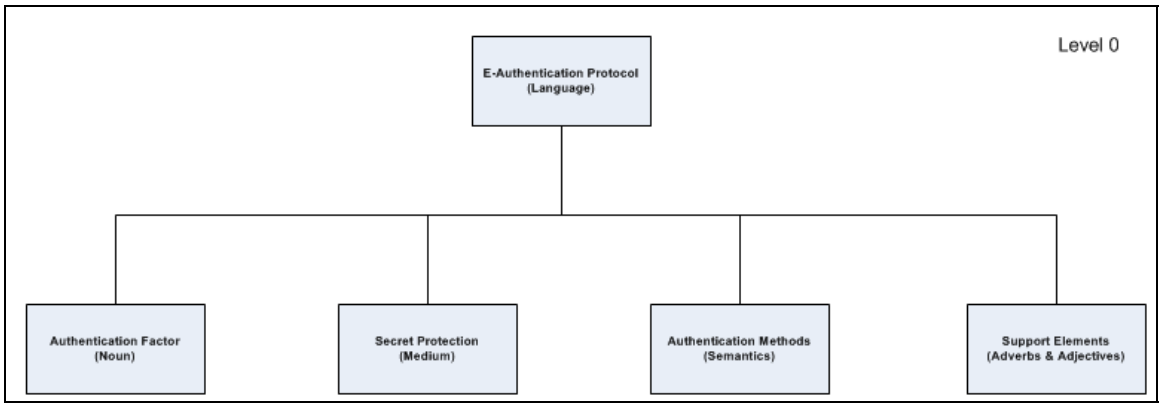


Figure 30. Overview of proposed taxonomy composition

### 1. Authentication Factor

The "Authentication Factor" classification criteria define the key secret component in an authentication process. The secret may exist as one or more of the authentication factors commonly categorized as what you know (a memorized secret), what you have (token based), and what you are (biometric based). The classification focuses on the knowledge and token based factors given this study's focus on e-authentication (online) protocols. Biometric based authentication factors are hence out of scope of the taxonomy.

To avoid possible confusion, we should distinguish the "what you know" form of PoP authentication from another form of authentication referred to as memory based authentication. With knowledge based authentication, the "knowledge" factor refers to some personal identification information; such as driver license number, or some other piece of personal information that is not generally known to anyone but you and a "trusted" third party (e.g., the DMV in

the case of your driver license number). Knowledge based authentication, however, is not based on true secrets, and is thus not considered further in this study.

Token based authentication factors entail some form of a physical "container" (e.g., smart card) that either contains secret values (e.g., key), symmetric or asymmetric, or houses a secretly-seeded algorithm that can generate a one-time secret. Asymmetric secrets refer to public and private key pairs. Non-repudiation protection is implicit with the implementation of asymmetric secret based authentication, given the "singularity" property required of all public-key cryptographic implementations. This property states that only the owner of a particular private key should ever have access to that key. The digital signing of a challenge resulting from the use of a private key thus allows for verification of the source of the signed challenge. Symmetric secrets can be static (e.g., a password) or dynamic (e.g., a one-time password).

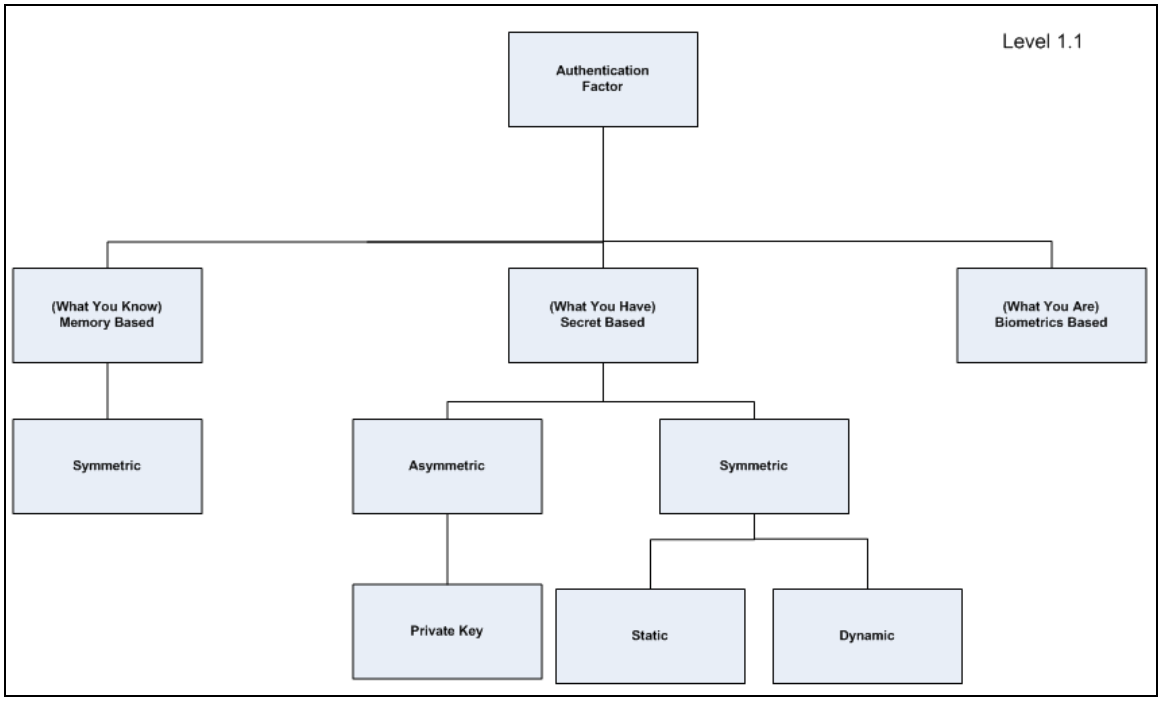


Figure 31. Classification based on Authentication Factor

### 2. Secret Protection

The classification of "Secret Protection" shows the various ways in which PoP of the secret can be conveyed in an authentication session. The secret can be transmitted in the clear (no protection), protected via symmetric or asymmetric cryptographic means, protected via hashing, or tunneled inside a secure communication channel (VPN) when one has been established.

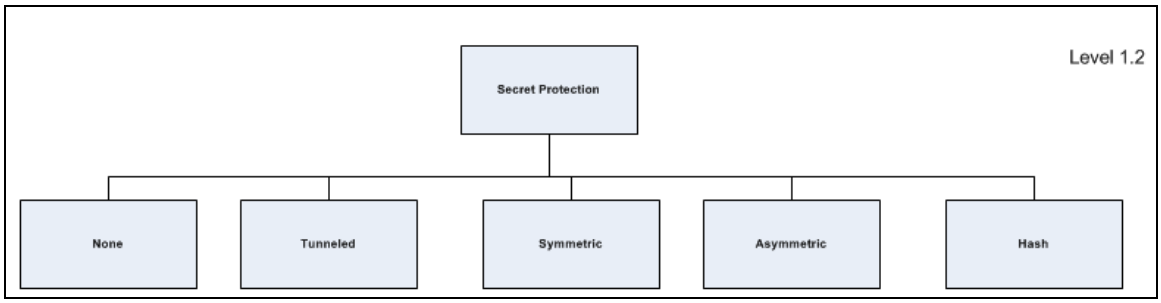


Figure 32. Classification based on Secret Protection

### 3. Authentication Methods

The "Authentication Methods" classification criteria define the various ways in which authentication is carried out to convey the PoP of a secret. The methods in which the authentication can take place may involve a direct presentation of the secret, or a series of message exchanges in terms of a challenge-response mechanism that precludes direct observation of the secret to an online observer. It may also be a zero-knowledge proof leveraging on mathematical algorithms and certain other properties decided on prior to deployment that not only precludes direct online observation of the secret, but also precludes a database attack on any of the relying party systems (i.e., no "secrets" are stored on these systems).

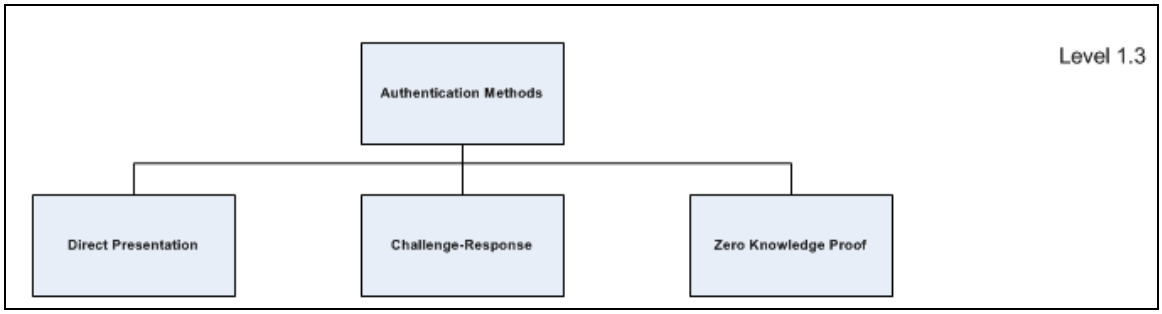


Figure 33. Classification based on Authentication Methods

### 4. Support Elements

The "Support Elements" classification recognizes the additional services that are provided by the authentication protocols in supporting data integrity and maintaining the state of the authenticated session. These support elements sometimes play critical roles in the authentication session regarding what security services are provided.

The existence and implementation of some support elements are crucial in addressing the authentication threats. Random numbers are a key element in providing data integrity, and also typical inputs to generating shared secret keys (i.e., session keys used to encrypt data exchanges after authentication has occurred). Nonces and timestamps are key elements used in the techniques that e-authentication protocols employ in attempting to address and protect against impersonation attacks that rely on the ability to replay certain critical authentication messages to the authenticator.

Maintaining session "identity" of an authentication session facilitates tracking the state of the session, resulting in a stateful protocol. A stateful protocol may be able to facilitate efficient session resumption in cases of timeout for an inactive session. As a result, the re-authentication process may be completed in fewer message exchanges as compared to having to complete a new authentication process from scratch. In contrast, a stateless protocol does

not maintain any information from the authenticated session, and thus any session resumption will need to be done with the complete message exchanges for the authentication process.

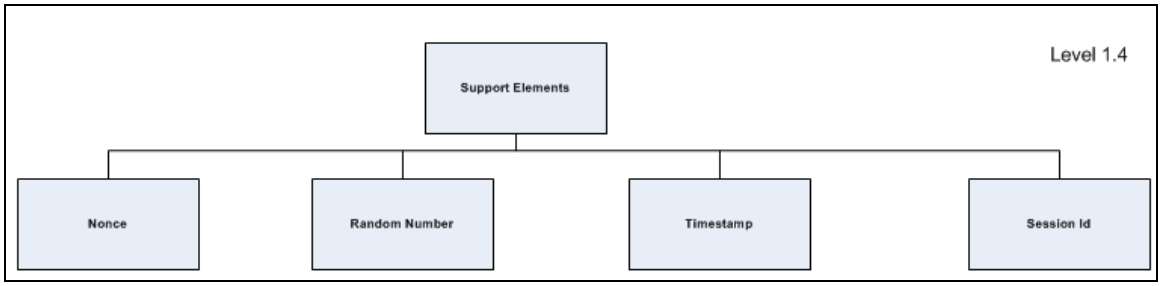


Figure 34. Classification based on Support Elements

## D. TAXONOMY ANALYSIS

In developing the taxonomy, it is possible to provide different perspectives of the characteristics of an e-authentication protocol. The taxonomy facilitates analysis of different aspects of the protocols, such as examining the strengths and weaknesses of the protocols given the mechanisms, key elements, and the secret used in the authentication process. This enables useful comparison and critical analysis of the capabilities that the protocol serves to provide.

As illustrated in Figure 35, the mechanisms in which the secret is exchanged during the authentication process can be organized and analyzed in another tree view to illustrate the protocol strengths and weaknesses. Protocols that are classified as employing the use of OTP are more secure and are protected against replay and brute force attacks. A static symmetric secret that is used in conjunction with a challenge-response mechanism adds dynamism to the authentication factor and is therefore not vulnerable to replay, though it may still be attacked by brute force. The use of a static symmetric secret via direct presentation is the weakest of all, as it is trivially subjected to replay and requires no brute force effort at all.

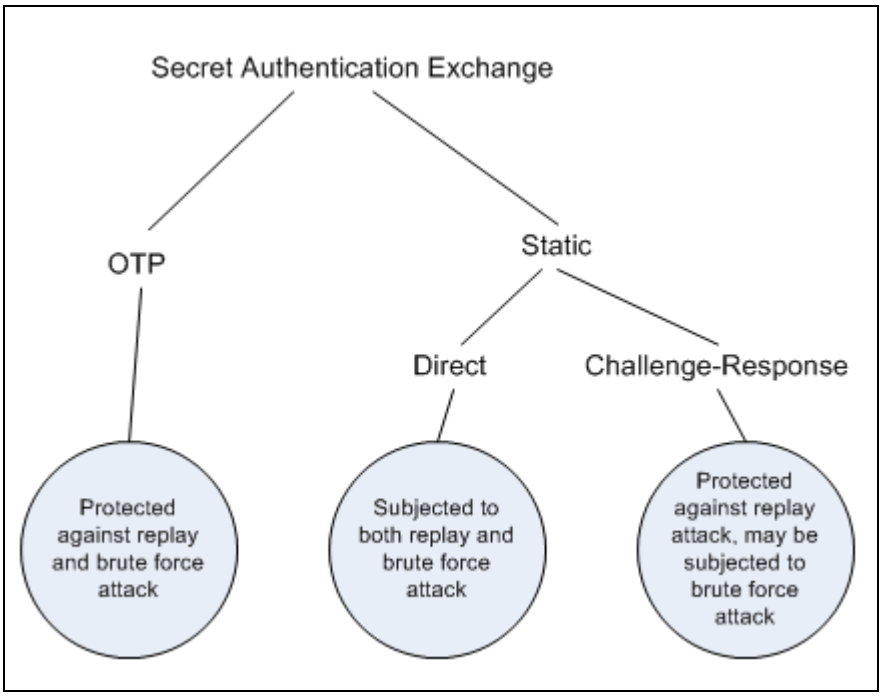


Figure 35. Another Tree View – Strengths and Weaknesses

## E. TAXONOMY APPLICABILITY

The proposed taxonomy for authentication protocols provides key components and elements to enable description of the protocol functionality, mechanisms, strengths, and weaknesses. This facilitates greater understanding of the authentication protocols in the selection of the appropriate protocols to address particular system requirements, operating environments, targeted threats, and risks. The evaluation of the protocols will be more apparent and more easily facilitated using the proposed taxonomy.

Based on the proposed taxonomy for authentication protocols, it is important to consider how the taxonomy can be validated, and how different perspectives can be created to facilitate selection of potential authentication protocols for the problem domain application. The following table shows one dimension of how the taxonomy can be leveraged to create a tabular view in categorizing the authentication protocols. This limited proof of concept illustrates

that each classification criterion is represented as a taxonomy tuple. In aggregating all the taxonomy tuples, it is possible to describe the basic authentication factor used, authentication mechanism, threats that are addressed, and whether the protocol is cryptographically protected.

| Taxonomy Tuples | | | | Notes | Candidate Authentication Protocols |
|---|---|---|---|---|---|
| Auth Factor | Auth Methods | Secret Protection | Supporting Elements | | |
| Password | Direct | None | - | - | PAP |
| Password | Challenge-Response | Hash | Nonce | Replay Protection | IPSEC-IKE |
| Public/Private Key Pair | Challenge-Response | Symmetric | Session Id | Session Resumption | SSL/TLS |
| Public/Private Key Pair | Challenge-Response | Symmetric | Random Number | Non-repudiation | SSL/TLS |
| Ticket Based | Direct | Symmetric | Timestamp | Replay Protection | Kerberos |
| Password | Zero Knowledge | Symmetric | - | Non-repudiation | SRP |
| Password | Direct | None | - | - | Telnet |
| Ticket Based | Direct | Symmetric | Timestamp | Replay Protection | Telnet using Kerberos authentication options |
| Public/Private Key Pair | Challenge-Response | Asymmetric | - | Non-repudiation | SSH |
| Password | Direct | Tunneled | - | Replay Protection | SSH |
| Password | Challenge-Response | Hash | - | - | CHAP, RADIUS, TACACCS+ |
| Password | Challenge-Response | Symmetric | - | - | Wireless Authentication, GSM Authentication, NTLM |
| Public/Private Key Pair | Challenge-Response | Asymmetric | Random Number | - | E-voting Authentication |
| Password | Challenge-Response | None | Random Number | - | Mifare Proprietary |

Table 1. Taxonomy Tuples Table

With the proposed taxonomy tree and table created, it might be interesting to see how this can be applied to solve real-world authentication problems. For example, a particular authentication requirement may require specific authentication factors and mitigation of specified threats. The taxonomy table can be applied to identify possible protocol candidates.

However, this limited taxonomy was focused on examining the authentication transactions between two parties. It may lack differentiating factors between the authentication protocols in terms of the setup required and

operating environment within which the protocol is to be deployed. In seeking a potential candidate protocol for use in a real world scenario, the authentication function setup, trusted third party, protocol overheads, key management infrastructure, and network infrastructure should also be considered.

### 1. Authentication Function Setup

The authentication function setup describes how the authentication is to work in the intended operating environment. The authentication function setup may exist in a centralized or distributed fashion. In a centralized setup, all Claimants will go through a single Verifier for authentication. This is similar to a typical client-server setup where the multiple clients will connect and authenticate to the single server.

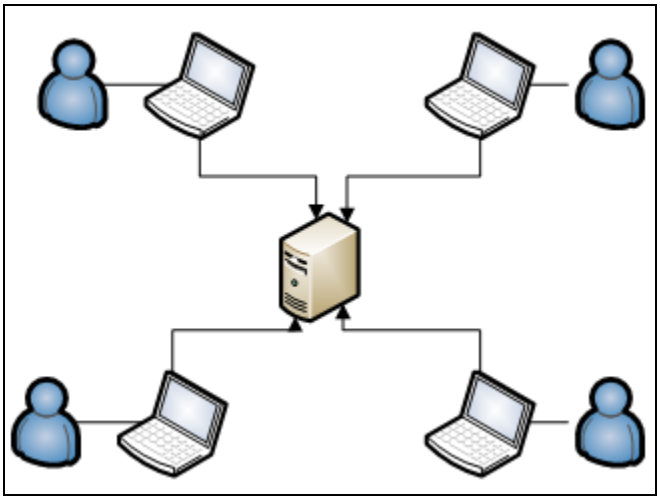

Figure 36. Centralized Authentication Function Setup

In a distributed setup, multiple Verifiers may exist, to which the Claimant can authenticate depending on the location, domain, proximity, or service functionality required. This means that the multiple Verifiers will have to maintain any requisite authentication information. Some authentication protocols support the structure of having only a *primary* Verifier responsible for maintaining the Claimant's authentication information. If the Claimant needs to utilize another Verifier for authentication, that (non-*primary*) Verifier may not have the

authentication information (secrets) necessary to ascertain the Claimant's identity. An assertion mechanism may facilitate communication between the two Verifiers in order to remedy this situation. Another mechanism is for the *Claimant* to deliver the assertion from the primary Verifier to the other Verifier for authentication.

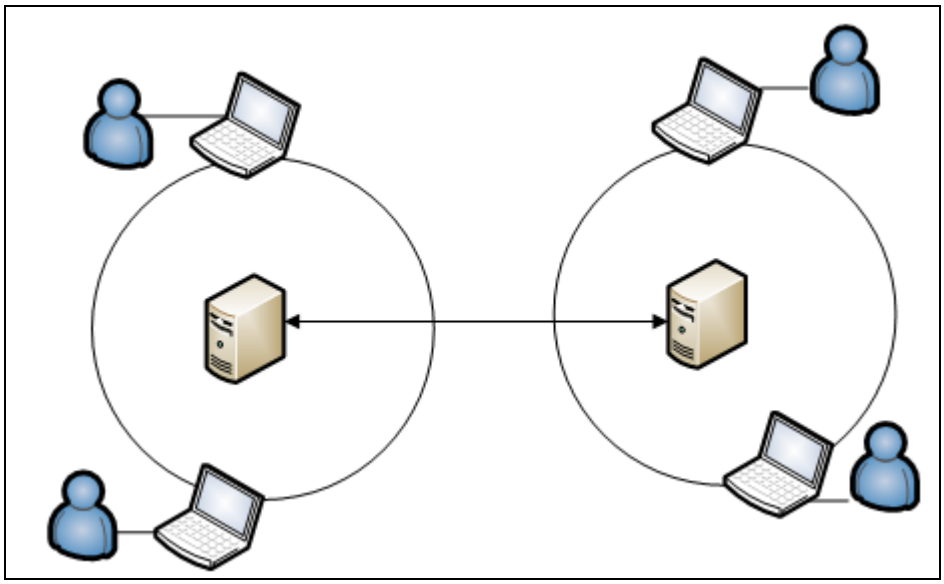

Figure 37. Distributed Authentication Function Setup (Multiple Verifiers)

### 2. Trusted Third Party

A trusted third party is an entity which is trusted by all parties participating in a given authentication protocol. Although trusted third parties may be an integral component and designed into some authentication protocols, this is not modeled in the proposed taxonomy, but rather assumed depending on the type of authentication involved (e.g., a Certification Authority in support of asymmetric-key authentication and a Key Distribution Center in support of Kerberos authentication). The reason for omitting this modeling is that any authentication transaction between a Relying Party and a Verifier is simply another instance of a Claimant-to-Relying Party authentication transaction.

However, this raises another issue; whether or not the authentication exchange is trust based or non-trust based. In a trust-based environment, the authenticating parties are within a circle of trust. As such, the respective authenticating parties make authentication decisions independently. In a non-trust based environment, usually a third party is required to act as the trusted authentication authority, or Verifier, to validate the Claimant's identity on request by a Relying Party.

### 3. Protocol Overheads

The protocols' overheads are another issue worthy of consideration. Protocol overhead refers to the size of the message transactions during an authentication session, the number of authentication messages required, the amount of processing required, and the amount of memory required. These will have a significant impact on the intended operating environment. If the handshaking process is long and involves several authentication message exchanges, this will not be ideal for a limited bandwidth network environment or for time critical systems.

Protocols that have a shorter handshaking process and support session resumption will be beneficial to deployment in a mobile environment or where the communication link's persistency may be intermittent. Authentication session resumption will enable the re-authentication process to be completed in fewer message exchanges as compared to the complete authentication process.

### 4. Support for Key Management Infrastructure

Secret management infrastructure refers to the processes and resources required for the management, control, and distribution of secrets. This includes the generation of secret keys, distribution of keys, renewal of keys upon expiry, as well as revocation of compromised keys. In some instances, the authentication protocol provides support for secret re-issuance, renewal, and

revocation. Attributes, such as a validity timestamp of the secret key, are verified to check for expired keys. The renewal or revocation processes may be initiated as part of the authentication protocol.

### 5. Network Infrastructure

Authentication protocols are designed to work in specific network conditions and environments. Minimum bandwidth requirements will need to be adhered to in order for the authentication protocols to work effectively. There are instances when a persistent TCP connection is required for the duration of not only the authentication protocol, but also the remainder of the transaction where the claimant is granted access to the requested resource. While certain authentication protocols may support intermittent joining and leaving the session without the need to authenticate repeatedly, others require re-authentication once the persistent connection is broken.

It is critical to understand the prerequisites of network infrastructure requirements in order for the authentication protocol to work within the targeted environment. For persistent and closed network environments, there will be no requirement for the authentication protocol to support intermittent joining and leaving the authenticated session. Single factor authentication using a symmetric secret may be sufficient for a closed environment where only legitimate and cleared users can physically access the network. For mobile and open network environments, support for mobile users joining and leaving the network is required. The authentication protocol will need to be able to support effective re-authentication or enable tolerance for a valid authenticated session. Other considerations include deploying multifactor authentication for more secure authentication means in light of the higher risk of loss and exposure of secrets in such operating environments.

THIS PAGE INTENTIONALLY LEFT BLANK

# V. SUMMARY AND CONCLUSIONS

## A. SUMMARY AND KEY OBSERVATIONS

In the process of conducting the study on the various e-authentication protocols and developing the protocol taxonomy, the primary focus was in examining the mechanisms and key elements facilitating the authentication process. There may be differences in how each protocol is implemented; however, after peeling the outer layers and inspecting the underlying mechanism, it was determined that the fundamental mechanisms governing the way in which secrets are exchanged in an authentication session were common in all protocols. Proof of possession of a secret is conducted via asymmetric or symmetric means. Shared symmetric secret is the more commonly used means due to its efficiency, relative simplicity, and lower cost of implementation. However, asymmetric secrets are necessary when non-repudiation is a required security service, and to support large-scale enterprises that are not conducive to dynamically establishing symmetric keys.

The basis of building the taxonomy is dependent on the application of the taxonomy. The approach used in this study's taxonomy development was to perform functional decomposition of the protocol in terms of the functionality it provides, the mechanisms it utilizes, and the key elements for facilitating the operation of protocol function. This enabled a breaking-down into the fundamental building blocks of what constitutes the fundamental authentication part of the protocol. The development of the taxonomy in this way enabled different perspectives and analyses of the protocols' capabilities and their applicability.

There are also observations of the protocol development trend where the later protocol versions tend to be able to support different modes of operation, providing value-added functionality beyond what a typical e-authentication

protocol does. There are other protocols that are based on some meta protocol framework. These pose critical taxonomy design considerations on how these factors should be treated and classified.

### 1. Protocol Development

The protocol development trend was observed to evolve towards more flexibility and the ability to support more options with the later version releases or newly developed protocols. The implementation caters to enabling multiple modes of operations, ability to support multiple cryptographic options, etc. Such protocols strive to provide an all-encompassing solution for catering to the various authentication needs.

Taking SSL/TLS as an example, the protocol is able to support multiple cryptographic options. It is within the initial handshaking protocol of the authentication process where negotiation is done on the choice of cryptographic options to use. It supports the use of either symmetric and asymmetric secrets for authentication. It also supports both one-way and mutual authentication setup, subject to the authentication requirements of the required system.

Another example, Kerberos V5, provides significant extensions in terms of functionality beyond those provided in V4. The motivation is no doubt to provide greater flexibility in the operating environments in which Kerberos can be deployed. The newer version allows the support of different encryption algorithms, whereas the previous version assumes DES as the encryption algorithm. Other extensions to the functionality include managing longer ticket lifetimes and enabling different realms to have different master secret keys.

This characteristic of enabling multiple modes and options for operation poses a challenge in developing the taxonomy. Any attempt to put such a protocol within a classification scheme has the tendency of falling into multiple categories. This leads to the thinking that the taxonomy classification does not seem normalized, and multiple paths of traversal in classification are possible for

a single protocol. This dilemma may be resolved by compliments of the taxonomy tuples table, which is able to support such protocol characteristics and provides multiple permutations of the possible protocol classifications.

### 2. Segregation of Authentication Protocol and Key Exchange Protocol

In the process of taxonomy development, it is imperative to be able to differentiate between a key exchange protocol and an authentication protocol, which are often mistaken for one another. This is due to the many currently available authentication protocols that provide both the key exchange and authentication functions within one protocol implementation. Being highly related and dependent on each other in an authentication process, there are merits in such implementation whereby the required message exchanges during an authentication process attempt to perform key exchange as well as authentication at the same time. However, in building the taxonomy for e-authentication protocol, it is imperative that focus be given to the authentication function and mechanism, rather than the key exchange protocol.

Meta protocols such as IPSec facilitate compliant protocols under their framework to be modular in fulfilling the specific objectives of performing initial endpoint authentication, session key generation/exchange, and subsequent data authentication and encryption. This allows the clear segregation of protocol functionality. Protocol replacement is then made easy in supporting upgrades to either of the component protocols, without affecting the overall behavior of the meta protocol.

### 3. Symmetric Key Distribution

Proof of possession of a symmetric shared secret is the most commonly used authentication means. The symmetric shared secret is either configured or derived within the authentication handshaking process. This is typically done via an agreed algorithm or mathematical properties. The question is whether is it

possible to derive a shared secret between authenticating parties without having pre-shared secrets distributed beforehand to serve as building blocks to generate the shared secret for authentication purposes and not for generation of a session key. From the study, it does not seem possible at this time, and none of the available protocols are able to accomplish that.

It is a well-accepted fact that symmetric shared secret based authentication is less costly to implement as compared to asymmetric means, owing to the fact that no complex key management infrastructure is required. However, the distribution of the symmetric keys remains the difficult issue to be addressed, and presents an ideal use case for asymmetric mechanisms (i.e., PKI) that effectively solve this distribution problem and that also provide the security objective of non-repudiation.

## B. RECOMMENDATIONS FOR FUTURE WORK

The proposed taxonomy is limited in its focus on authentication mechanisms only. There are other worthy considerations to extend the taxonomy to enhance differentiation between the authentication protocols for effective selection of a potential candidate protocol for the desired problem domain. This will require study beyond the authentication mechanism and relates to understanding the characteristics of the operating environment. Certain authentication mechanisms may prove to be more effective depending on the characteristics of the operating environment.

The study of protocol overheads is one key area in differentiating the authentication protocols in terms of their efficiency to complete the handshaking and authentication process within a certain number of messages, and in considering whether the required message size is reasonable for the constrained bandwidth of the operating environment. Quick authentication modes with a session resumption process would be beneficial for environments where the communication link is intermittent and unstable, or the Claimant is highly mobile.

The other notable issue is with regard to the authentication function setup supported. It would be appropriate for a highly mobile environment to require the authentication function setup to be distributed rather than centralized. A particular authentication protocol may satisfy the system requirement in terms of functionality but does not support distributed authentication setup. In another scenario, an authentication protocol may need to employ the services of a trusted third party and will have specific operating network environment requirements. The authentication protocol will not be effective if requirements are not as expected in the actual deployment environment.

Lastly, the support for a secret management infrastructure may be provided by some authentication protocol to incorporate the revocation process after assessing the token's validity. It may be a redundant feature if is not available for implementation in the actual operating environment.

In general, understanding the characteristics of the operating environment will place more demands on studying the authentication protocols from different perspectives and goes beyond the functionality and mechanism within the authentication process. This will indeed pose significant challenges to the extension and design of the taxonomy. However, a successful attempt in putting these considerations together to enrich the taxonomy will result in a more comprehensive and applicable taxonomy for addressing real deployment in the target problem domain.

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF REFERENCES


Aboba, B., L. Blunk, J. Volltrecht, J. Carlson, and Ed H. Levkowetz. *Extensible Authentication Protocol (EAP), RFC 3748.* June 2004.

"Authentication Protocol." *Wikipedia.* July 5, 2009. http://en.wikipedia.org/wiki/Authentication_protocol (accessed August 17, 2009).

Authentify Technology. *Authentify — Out of Band Multifactor Authentication.* 2009. http://www.authentify.com/solutions/outofband.html (accessed September 11, 2009).

Bingemann, Mitchell. *Centrelink Goes Biometric, Australian IT.* May 26, 2009. http://www.australianit.news.com.au/story/0,,25538088-15319,00.html (accessed August 17, 2009).

Chen, Xixi. "Xixi Chen's Homepage - Authentication Protocols in GSM Networks." *The VLSI Group at the University of Waterloo.* November 14, 2002. http://www.vlsi.uwaterloo.ca/~xxchen/ (accessed September 10, 2009).

Clark, John, and Jeremy Jacob. "A Survey of Authentication Protocol Literature: Version 1.0." November 17, 1997. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.57.3372 (accessed August 20, 2009).

Gafurov, Davrondzhon. "A Survey of Biometric Gait Recognition: Approaches, Security and Challenges." *Norwegian Informatics Conference (NIK).* Norway, 2007.

Gans, Gerhard de Koning, Jaap-Henk Hoepman, and Favio D. Garcia. *A Practical Attack on the MIFARE Classic.* 2008.

Goldberg, Ian. "The Insecurity of 802.11: An Analysis of Wired Equivalent Privacy Protocol." July 11, 2001. http://www.cypherpunks.ca/bh2001/mgp00001.html (accessed September 18, 2009).

*GSM Security FAQ.* 2003. http://www.gsm-security.net/gsm-security-faq.shtml (accessed September 9, 2009).

Hill, Joshua. "An Analysis of RADIUS Authentication Protocol." November 24, 2001. http://www.untruth.org/~josh/security/radius/radius-auth.html (accessed March 12, 2009).

Imperial College London. *Internet Banking Security.* http://mipagina.cantv.net/lumejr/developing.html (accessed September 11, 2009).

Jain, Anil K. "Biometrics: A Tool for Information Security." *IEEE Transactions on Information and Security* 1, no. 2 (Jun 2006): 125–143.

"JANET Technical Sheets." *JANET — The UK's Education and Research Network.* 2007. www.ja.net/documents/publications/factsheets/065-eap.pdf (accessed August 12, 2009).

Kaufman, Charlie, Radia Perlman, and Mike Speciner. *Network Security - Private Communication in a Public World.* New Jersey: Prentice Hall, 2002.

Kessler, Gary C. *An Overview of Cryptography.* June 4, 2009. http://www.garykessler.net/library/crypto.html (accessed August 11, 2009).

Krawczyk, Hugo. *SKEME: A Versatile Secure Key Exchange Mechanism for Internet.* 1996.

Lancaster, Tom. "Simple VPN Authentication Choices." *Search Enterprise Desktop.com.* March 13, 2002. http://searchenterprisedesktop.techtarget.com/tip/0,289483,sid192_gci997892,00.html (accessed September 10, 2009).

Martin, Luther. "Multi-Factor Authentication." *Voltage Superconductor.* March 30, 2009. http://superconductor.voltage.com/2009/03/multifactor-authentication.html (accessed August 17, 2009).

Maughan, D., M. Schertler, M. Schneider, and J. Tuner. *Internet Security Assocication and Key Management Protocol (ISAKMP), RFC 2408.* November 1998.

Meng, Bo. "A Secure Non-Interactive Deniable Authentication Protocol with Strong Deniability Based on Discrete Logarithm Problem and Its Application on Internet Voting Protocol." 2009. http://scialert.net/qredirect.php?doi=itj.2009.302.309&linkid=pdf.

NIST. *Electronic Authentication Guideline.* Information Security NIST Special Publication 800-series, NIST, 2008.

Orman, H. *The OAKLEY Key Determination Protocol, RFC 2412.* November 1998.

P. Karn, W. Simpson. *Photuris: Session Key Management Protocol, RFC 2522.* March 1999.

Patil, Sandeep Ramesh. "Multi-security mechanisms with multifactor authentications ." *IBM developerWorks.* March 10, 2009. http://www.ibm.com/developerworks/aix/library/au-security_auth/index.html (accessed August 17, 2009).

Rigney, C., S. Williams, A. Rubens, and W. Simpson. *Remote Authentication Dial In User Service (RADIUS), RFC 2865.* June 2000.

*SearchSecurity.com.* June 4, 2007. http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci211621,00.html (accessed August 9, 2009).

Simpson, W. *PPP Challenge Handshake Authentication Protocol (CHAP), RFC 1994.* August 1996.

Szabo, Nick. "Trusted Third Parties Are Security Holes." 2005. http://szabo.best.vwh.net/ttps.html (accessed August 17, 2009).

Thakker, Chetan. *WLAN Authentication Using EAP-SIM.* August 26, 2004. http://tcn.cse.fau.edu/asf/presentations/docs/chetan_WLanAuthenticationUsingEAP-SIM.ppt (accessed September 10, 2009).

Todorov, Dobromir. *Mechanics of User Identification and Authentication, Fundamentals of Identity Management.* New York: Auerbach Publications, 2007.

Wayman, James L. "Biometrics in Identity Management Systems." *IEEE Security and Privacy* 6, no. 2 (Mar/Apr 2008): 30-37.

Wu, T. *The SRP Authentication & Key Exchange System, RFC 2945.* September 2000.

Wu, Tom. "The Secure Remote Password Protocol." *Standard University.* November 22, 1997. http://srp.stanford.edu/doc.html#papers (accessed September 3, 2009).

Ylonen, T., and Ed. C. Lonvick. *The Secure Shell (SSH) Protocol Architecture, RFC 4251.* January 2006.

Zorn, G. *Microsoft PPP CHAP Extensions, Version 2, RFC 2759.* January 2000.

# INITIAL DISTRIBUTION LIST

1. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

2. Dr Peter Denning
   Naval Postgraduate School
   Monterey, California

3. Dr Ted Huffmire
   Naval Postgraduate School
   Monterey, California

4. Mr J. D. Fulp
   Naval Postgraduate School
   Monterey, California

5. Professor Yeo Tat Soon, Director
   Temasek Defence Systems Institute
   National University of Singapore
   Republic of Singapore

6. Ms Tan Lai Poh, Assistant Manager
   Temasek Defence Systems Institute
   National University of Singapore
   Republic of Singapore

7. Ms Chia Wan Yin
   Defence Science & Technology Agency
   Republic of Singapore